Poszukiwanie prostej i uniwersalnej maszyny obliczeniowej

Algorytm = operacje elementarne + struktury sterujące

Program obliczeniowy =
= język programowania + algorytm + struktury danych

Maszyna obliczeniowa (**MO**) =
= model pamięci + zapis sterowania + zestaw operacji elementarnych

Jak dalece można uprościć <u>model pamięci</u>, aby można było tworzyć różne struktury danych?

Czy pamięć MO mogłaby być jednowymiarową „taśmą” podzieloną na komórki?

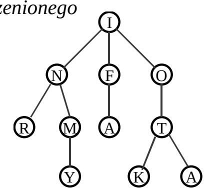
Przykład tablicy dwuwymiarowej

7	45	–3
91	0	12
–15	11	17

#	7	45	–3	∗	91	0	12	∗	-15	11	17	#

Strukturę tablicy można zapisać na jednowymiarowej taśmie

Przykład drzewa ukorzenionego



#	**I**	∗	**N**	**F**	**O**	∗	**R**	**M**	**A**	**T**	∗	**Y**	**K**	**A**	#

lub lepiej

(**I**) (**N** **F** **O**) (**R** **M**) (**A**) (**T**) ( ) (**Y**) ( ) (**K** **A**)

Strukturę drzewa można zapisać na jednowymiarowej taśmie

Każdą strukturę danych da się **zlinearyzować**, tzn. zapisać na jednowymiarowej taśmie

W MO przyjmujemy najprostszy model pamięci:

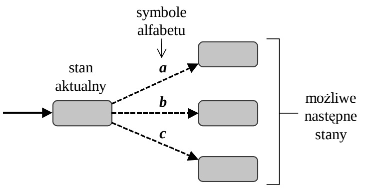
		#	#														#	#		

- pamięcią MO jest <u>nieskończoną</u> jednowymiarową taśmą podzieloną na komórki,
- pojedyncza komórka może przechowywać jeden znak z przyjętego alfabetu symboli,
- puste komórki wypełnione są umownym symbolem **#**

Jak dalece można uprościć <u>zapis sterowania</u>, tak aby można było realizować różne algorytmy?

Sterowanie powinno wskazywać kolejność wykonywania operacji w procesie wyznaczania wyniku wg przyjętego algorytmu (w każdym momencie może być wykonywana co najwyżej jedna operacja)

- znajdowanie się w określonym miejscu procesu realizacji algorytmu nazwiemy **stanem** procesu,
- stan początkowy symbolizuje rozpoczęcie realizacji algorytmu (*start*), a stan końcowy jego zatrzymanie (*stop*),
- proces jest sekwencją stanów przechodzonych jeden po drugim,
- w każdym przebiegu algorytmu kolejność stanów wynika z zawartości określonych komórek w pamięci MO.



- wszystkie stany, które mogą pojawić się przy realizacji danego algorytmu są zaznaczane jako węzły na tzw. **diagramie przejść międzystanowych**
- dla każdego stanu na diagramie, zbiór następnych stanów wynika ze struktury sterowania przyjętej w danym algorytmie
- wybór jednego z możliwych stanów następnych zależy od zawartości komórek pamięci, w których zapisano dane początkowe dla konkretnego przebiegu algorytmu

Jak dalece można uprościć zestaw operacji, tak aby można było realizować różne algorytmy?

Zestaw podstawowych operacji musi zawierać zapisanie i odczytanie symbolu z komórki pamięci, wskazanie pojedynczej komórki w celu dokonania zapisu lub odczytu oraz przejście do kolejnego stanu.

Okazuje się, że nie musi zawierać już niczego więcej!

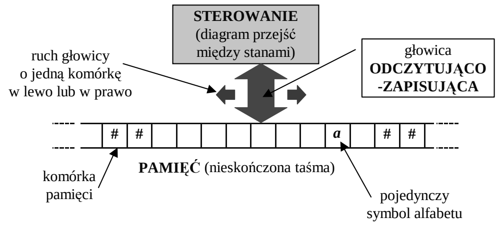
Operacje wykonywane przez MO:

1. przeczytanie jednego symbolu ze wskazanej komórki taśmy,
2. zapisanie jednego symbolu do wskazanej komórki taśmy,
3. wskazanie jednej z dwóch komórek sąsiednich po wykonaniu odczytu i zapisu w bieżącej komórce,
4. przejście do kolejnego stanu.

Wszystkie podane operacje mogą być wykonywane zawsze razem

Maszyna Turinga (MT)

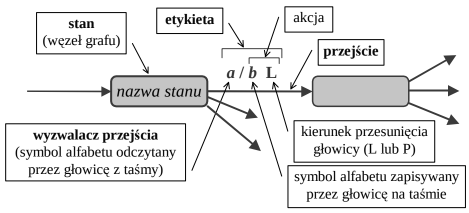
Jeden z wielu modeli prostej MO



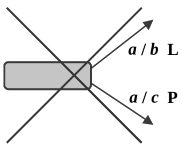
Na **MT** składa się:

- skończony **alfabet** symboli do zapisywania danych,
- skończony zbiór **stanów**, w których może się znajdować proces realizacji algorytmu zapisanego w postaci MT,
- nieskończona **taśma** podzielona na **komórki** przechowujące pojedyncze symbole alfabetu,
- krokowo poruszająca się **głowica** odcztująco-zapisująca,
- **diagram przejść** miedzy stanami, który steruje głowicą tak, że po każdym odczytaniu zawartości komórki następuje zapisanie do niej podanego symbolu, głowica jest przesuwana w lewo lub w prawo o jedną komórkę i następuje przejście do kolejnego stanu, co rozpoczyna znów tę samą sekwencję operacji,
- jeden stan **początkowy** i stany **końcowe**, których może być kilka.

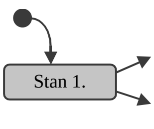
Węzłami (wierzchołkami) diagramu przejść są wszystkie stany w jakich może znajdować się MT podczas realizacji algorytmu, dla którego została zdefiniowana.
Węzły na diagramie są połączone odcinkami skierowanymi, które symbolizują możliwe przejścia od stanu do stanu.
Każde przejście opisane jest tzw. **etykietą**.



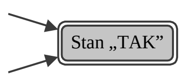
- dla każdego stanu w diagramie MT muszą być opisane przejścia do kolejnych stanów dla wszystkich symboli w przyjętym alfabecie i dla dodatkowego symbolu #,
- MT jest **deterministyczna,** jeżeli z żadnego stanu nie wychodzi więcej niż jedno przejście z tym samym wyzwalaczem, tzn. nie pozostawiamy wyboru przejścia maszynie, zakładając, że wybierze właściwie; odczytanie z taśmy symbolu wyzwalacza jednoznacznie wskazuje, (determinuje), który ze stanów jest stanem następnym,



- dokładnie jeden ze stanów w diagramie przejść MT jest wyróżniony jako **stan początkowy**,

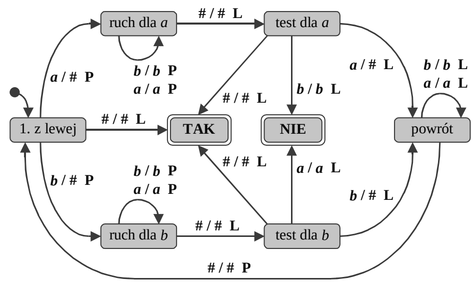


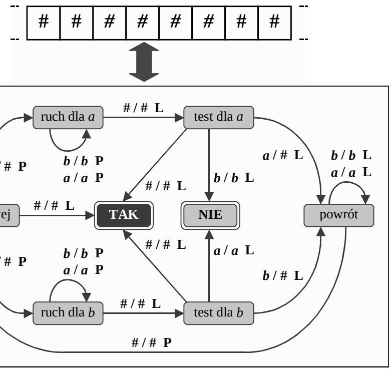
- w stanie początkowym głowica MT jest ustawiona na pierwszej od lewej niepustej komórce taśmy,
- stany, z których nie wychodzą już żadne przejścia, nazywane są **stanami końcowymi** – MT kończy w nich swoje działanie,



Przykład MT do wykrywania palindromów

Alfabetem symboli jest zbiór {*a*, *b*}





Przykład taśmy MT przy dodawaniu dwóch liczb dziesiętnych

#	#	#	#	#	#	#	#	7	3	6	+	6	3	5	1	9	#
#	#	#	#	#	#	5	!	7	3	#	+	6	3	5	1	#	#
#	#	#	#	#	5	5	!	7	#	#	+	6	3	5	#	#	#
#	#	#	#	2	5	5	!	#	#	#	+	6	3	#	#	#	#
#	#	#	4	2	5	5	!	#	#	#	+	6	#	#	#	#	#
#	#	6	4	2	5	5	!	#	#	#	+	#	#	#	#	#	#
#	!	6	4	2	5	5	!	#	#	#	+	#	#	#	#	#	#

MT może realizować także algorytmy obliczeniowe

Jakie algorytmy może realizować MT?

Wszystkie!

Teza Churcha-Turinga (tzw. Teza CT)

Na maszynie Turinga można zrealizować rozwiązanie każdego efektywnie rozwiązywalnego problemu algorytmicznego!

MT jest uniwersalną MO

efektywnie rozwiązywalny problem algorytmiczny = problem, dla którego można znaleźć algorytm dający się zapisać w jakimkolwiek języku jako program wykonujący się na jakimkolwiek istniejącym lub możliwym do zbudowania komputerze (nawet jeśli wymagałby on nieograniczonej ilości czasu i pamięci)

MT stosowana jest w wielu odmianach:

- maszyna z taśmą tylko jednostronnie nieskończoną,
- maszyna z wieloma taśmami i głowicami,
- maszyna z taśmą dwuwymiarową,
- maszyna niedeterministyczna.

Teza CT

Wszystkie te odmiany są **równoważne** w sensie zbioru problemów algorytmicznych, które mogą być nimi rozwiązane.

Proponowano różne inne uniwersalne MO:

rachunek lambda (A. Church), system produkcji do manipulowania symbolami (E. Post), klasa funkcji rekurencyjnych (S. Kleen) itd.

Wszystkie te modele są **równoważne** MT ⇔ **Teza CT**