

Streszczenie popularnonaukowe Klasycznym pojęciem obliczeń “wydajnych” w informatyce jest klasa PTIME, a więc te programy, których czas działania ograniczony jest wielomianem, jak np. $3n^2 + 7n + 5$, gdzie n jest rozmiarem wejścia. Celem projektu jest stworzenie odpowiednika tej teorii, ale dla model skończenie stanowych, takich jak automaty skończone. Chcemy badać przekształcenia, które operują na obiektach takich jak listy czy drzewa, czynią to w sposób skończenie stanowy i grają podobną rolę dla automatów jak klasa PTIME dla obliczeń ogólnych. Oprócz ciekawości poznawczej, modele skończenie stanowe będziemy badać dlatego, że mają rozstrzygalny problem stopu, w odróżnieniu od ogólnych maszyn Turinga.

Czym są właściwie przekształcenia skończenie stanowe? Choć pełnej odpowiedzi nie znamy – a celem projektu jest jej poszukiwanie – to odpowiedź znamy w przypadku *języków*, czyli funkcji których wejścia są słowami, a wyjściami są wartości boolowskie “tak” i “nie”. W przypadku języków, za skończenie stanowe uznaje się tzw. języki regularne. Teoria języków regularnych jest stabilna i w projekcie nie zamierzamy jej dalej rozwijać. Jednym z jej uroków jest mnogość równoważnych definicji, w tym: automaty skończone (dziesiątki równoważnych modeli), wyrażenia regularne, półgrupy skończone, czy też różnego rodzaju logiki. Każda z tych definicji ma swoje zastosowania, a ich równoważność jest zjawiskiem niezwykle i stanowi o atrakcyjność teorii. Teoria ta jest badana od samego początku informatyki, a jednym z ważnych jej nurtów są motywacje z dziedziny logiki. Badania te miały olbrzymi wpływ na informatykę, nie tylko teoretyczną, o czym świadczą cztery nagrody Turinga: Rabin i Scott (1976) za wprowadzenie automatów niedeterministycznych; Pnueli (1996) za użycie logiki temporalnej dla opisu poprawności programów (obecnie logikę temporalną bada się głównie za pomocą automatów skończonych); Clarke, Emerson i Sifakis (2007) za modelowanie programów i protokołów za pomocą automatów skończonych; oraz Aho i Ullman (2020) za prace nad implementacją języków programowania, w których ważną rolę grają automaty używane przy parsowaniu.

Tradycyjnie automat dostaje na wejściu słowo, a na wyjściu zwraca wartość boolowską. Poza budynkiem uczelni, programy jednak nie ograniczają się do odpowiedzi “tak” czy “nie”. Mogą przeformatować plik, czy też wykonać inny program. Dla wielu modeli obliczeń – jak np. w ogólnych maszynach Turinga, czy też w maszynach czasu wielomianowego – nie ma większej różnicy między wyjściami boolowskim a bardziej złożonymi, gdyż drugie można zastąpić pierwszymi (np. rozważając pytanie “czy, jeśli wejściem jest słowo w , to n -tym bitem wyjścia jest litera a ?”). Dla automatów skończonych sztuczka ta nie działa, dlatego też w przypadku automatów skończonych, funkcje o wyjściach ogólnych (zwane transdukcjami), mają osobną i bogatszą teorię niż funkcje o wyjściach boolowskich (czyli języki). W szczególności dla transdukcji zaczyna mieć sens ograniczenie wielomianowe, rozumiane jako ograniczenie na rozmiar wyjścia.

Transdukcje badano od początku teorii automatów; można nawet powiedzieć, że pierwsze historycznie modele bliższe były transdukcjom niż językom. W swojej sławnej pracy, Rabin i Scott zaproponowali odejście od języków, pisząc, że “we are doing away with a complicated output function and having our machines simply give *yes* or *no* answers”. Ich śladem, większość późniejszych badań w teorii automatów dotyczyło języków a nie transdukcji. Projekt ten za przedmiot badań obiera jednak właśnie transdukcje.

Wczesne modele transdukcji były prostymi wariantami automatów, np. maszyna Mealyego to jednokierunkowy automat deterministyczny, który przy każdej czytanej literze wejściowej produkuje jedną literę wyjściową. Z czasem zaczęły pojawiać się modele ciekawsze i trudniejsze; a współczesne modele są już wyposażone w takie konstrukcje programistyczne jak zmienne, pętle, rekurencja, czy funkcje wyższego rzędu. Oznacza to, że mogą być wykorzystywane jako rzeczywiste programy, a nie tylko jako wyidealizowane i radykalnie uproszczone modele tychże programów. Wszystko to przy zachowaniu ograniczeń skończenie stanowych, dzięki którym problem stopu pozostaje rozstrzygalnym (co jest w kontekście tego projektu rozumiane jako zaleta).

Podstawowym celem niniejszego projektu jest

Wykształcenie pojęcia wielomianowych obliczeń skończenie stanowych.

Wcześniejsze prace wykształciły już takie pojęcie dla szczególnego przypadku liniowego, lecz przypadek wielomianowy pozostaje w dużej mierze niezbadany. Zamierzamy: (a) zbadać jakie struktury danych mogą być przedmiotem obliczeń skończenie stanowych; (b) zaproponować modele wielomianowe, które operują na tych strukturach danych; (c) opracować algorytmy, które rozstrzygają własność zaproponowanych modeli; oraz (d) udowodnić, że zaproponowane modele są jedynymi możliwymi, czyli istnieje tylko jedna odpowiedź na rozważane pytanie.