Automaty można postrzegać jako modele systemów (programów) komputerowych. Czym jest model? Ogólnie rzecz ujmując, jest to uproszczony obraz rzeczywistości, który pomaga zobrazować jakiś problem. Modele wykorzystywane są w wielu dziedzinach: mamy modele samolotów, modele osobowości, modele wszechświata itp. Na modelach pracują przedstawiciele wielu dziedzin nauki; również informatycy.

Celem niniejszego projektu jest rozwiązanie niektórych problemów klasycznej teorii automatów, które można nazwać „zapomnianymi problemami". Jakiś czas temu (np. 10, 20 czy 50 lat temu), uznawano je za ważne, fundamentalne problemy. Były one jednak na tyle trudne, że nikt nie mógł ich rozwiązać. Jednocześnie udało się poczynić postępy w teorii w innych kierunkach, nie rozwiązując tych problemów. Z tych powodów ludzie przestali nad nimi aktywnie pracować.

Zamierzamy przywołać te problemy do życia i jeszcze raz spróbować je rozwiązać. Naszą nadzieję bazujemy na fakcie, że w międzyczasie powstało wiele interesujących narzędzi (twierdzeń). Tak więc, chociaż problemy te były zbyt trudne dla osób próbujących je rozwiązać te, powiedzmy, 20 lat temu, mogą nie być już takie trudne obecnie, gdy mamy dostęp do dużej ilości dodatkowej wiedzy.

Dopowiedzmy, że z jednej strony zarówno teoria, jak i praktyka rozwijają się dobrze, pomimo że wyżej wymienione problemy nie są rozwiązane. Z drugiej strony problemy te dotyczą podstawowych właściwości powszechnie używanych obiektów, więc byłoby wspaniale je rozwiązać.

Dopowiedzmy również, że projekt wpisuje się w badania podstawowe; chcemy teoretycznie zrozumieć i wyjaśnić pewne zjawiska. Nie spodziewamy się natychmiastowych praktycznych, komercyjnych zastosowań.

Opiszemy teraz nieco bardziej konkretnie problemy, które chcemy rozwiązać. Projekt obejmuje cztery główne zadania badawcze:

Upraszczanie automatów. Pierwsza grupa problemów dotyczy pytań następującej postaci: podać algorytm, który mając dany jakiś bardziej skomplikowany automat (np. automat ze stosem) stwierdza, czy istnieje prostszy automat robiący to samo (np. automat skończony rozpoznający ten sam język) i znajduje go, jeśli istnieje. Problemy takie można rozpatrywać dla różnych rodzajów automatów. Chcemy podać takie algorytmy dla najczęściej używanych typów automatów. Najczęściej albo żaden algorytm nie jest znany, albo mamy tylko bardzo wolne algorytmy. Dodajmy jednak, że zaprojektowanie algorytmów nie jest najważniejszym celem; prawdziwym (ukrytym) celem jest dogłębne zrozumienie, które automaty można faktycznie zmienić na prostsze.

Problemy równoważności. W problemach z tej grupy celem jest podanie algorytmu, który dostając dwa automaty (tego samego rodzaju) stwierdza, czy są one równoważne. Oczywiście aby to pytanie miało sens, musimy sprecyzować, co oznacza „równoważny"; istnieje wiele rozsądnych opcji do wyboru, ale generalnie mamy tu na myśli raczej zachowanie automatów niż ich wygląd. I to jest źródło trudności: algorytm może otrzymać dwa automaty, które wyglądają zupełnie inaczej, chociaż w rzeczywistości robią to samo. W projekcie rozważamy pytania tego rodzaju dla automatów ze stosem (reprezentujących programy rekurencyjne), a także dla wyrażeń rachunku π (reprezentujących komunikujące się procesy współbieżne). Ponownie, w zależności od konkretnego przypadku, albo nie jest znany żaden algorytm, albo mamy tylko niezwykle wolne algorytmy.

Wyrażalność automatów ze stosem wyższego rzędu. Automaty te zostały wprowadzone jako model programów korzystających z rekurencji wyższego rzędu (tj. rekurencji, w której funkcje mogą przyjmować inne funkcje jako parametry; rekurencja taka pojawia się w większości współczesnych języków programowania). Mimo że automaty te okazały się bardzo przydatne, pewne podstawowe zagadnienia związane z ich wyrażalnością pozostają nierozwiązane. Na przykład istnieją dwa warianty tych automatów: z operacją paniki (ang. *collapse*) i bez tej operacji; chcemy udowodnić, że wariant z paniką, który wydaje się silniejszy, faktycznie może poradzić sobie z jakimś przykładem, którego (dowodliwie) nie da się obsłużyć automatem bez paniki. Kolejne pytanie brzmi: jak te automaty mają się do gramatyk kontekstowych, jednego z czterech podstawowych typów gramatyk wprowadzonych przez Chomsky'ego w 1956 roku?

Automaty na nieskończonych słowach i drzewach. Automaty działające na nieskończonych słowach i drzewach mogą być używane jako modele programów, które działają w nieskończoność (np. serwery – nie są to programy, które mają coś obliczyć i zatrzymać się w skończonym czasie). Podczas gdy słowa pozwalają na zakodowanie pojedynczego obliczenia, rozgałęziający się charakter drzew pozwala na zakodowanie wszystkich możliwych obliczeń programu na raz. Takie automaty działające przez nieskończony czas, zwłaszcza te operujące na drzewach, nie są jeszcze dobrze poznane. Chcemy zrozumieć (i pokazać rozstrzygalność) kilku podstawowych problemów dotyczących tych automatów.