Matematyka jest królową nauk. Inne nauki, takie jak fizyka czy astronomia, opierają się na obserwacjach i doświadczeniach żeby osiągnąć akceptowalny poziom pewności, tymczasem matematycy są w stanie osiągnąć absolutną pewność dowodząc swoich twierdzeń. Pewność matematyczna jest najwyższym stopniem pewności, jaki ludzkość kiedykolwiek osiągnęła. Twierdzenie Pitagorasa ma już około 2500 lat i nigdy nie było podważane.

Dowody matematyczne to rozumowania składające się z małych kroków, które każdy może sprawdzić. Są one publikowane w artykułach naukowych lub książkach, które każdy może przeczytać i sprawdzić ich poprawność. Zupełnie inaczej jest w naukach doświadczalnych, gdzie powtórzenie eksperymentu zwykle wymaga użycia drogiego sprzętu nieosiągalnego dla zwykłych ludzi.

Niestety w ostatnich latach sytuacja zaczęła się zmieniać, ponieważ matematyka bardzo się skomplikowała. Dowody twierdzeń stały się tak duże i skomplikowane, że pojedyncza osoba nie jest w stanie sprawdzić ich w całości. Niektóre dowody opierają się na innych dowodach napisanych przez kogoś innego. Różne części dowodu mogą opierać się na różnych dziedzinach matematyki. Często matematycy biorą wyniki z książek jako punkty startowe do swojej własnej pracy. We wszystkich tych przypadkach mamy do czynienia z dowodami tak skomplikowanymi, że jedna osoba nie może zapanować nad ich całością.

Rozwiązanie tego problemu może dać informatyka. Dowody matematyczne można pisać w języku zrozumiałym dla komputerów i przy pomocy komputerów sprawdzać. Istnieją już języki, w których można to zrobić. Powstały one po wielkim kryzysie w podstawach matematyki na początku XX wieku. Kryzys ten był spowodowany faktem, że matematycy zaczęli rozważać coraz większe i bardziej abstrakcyjne struktury, takie jak nieskończone zbiory, nieskończone zbiory nieskończonych zbiorów, zbiór wszystkich nieskończonych zbiorów itp. Niektóre z tych struktur stały się tak abstrakcyjne, że wielu matematyków nie wierzyło w ich istnienie, a kiedy coś nie istnieje, założenie istnienia tego czegoś jest fałszem. Ponieważ w matematyce z fałszu wynika wszystko, argumentowano, że być może cała ta abstrakcyjna matematyka została wyprowadzona z fałszu i jest rozważaniem o niczym.

Kryzys z początku XX wieku został zażegnany przez wprowadzenie aksjomatycznej teorii mnogości w latach 1930-1940. Definicje dużych obiektów zostały spisane w języku teorii mnogości i matematycy mogli wreszcie zająć się tym co lubią najbardziej, czyli dowodzeniem twierdzeń o tych obiektach. To oznacza, że od ponad 70 lat znane są języki i systemy logiczne, w których dowody matematyczne mogą być formalnie sprawdzone. Co więc powstrzymuje nas od używania tych języków?

Problem polega na tym, że matematycy, nawet gdy bardzo precyzyjnie wyrażają swe myśli, nie są wystarczająco precyzyjni dla komputera. Często używają oni argumentów typu "od razu widać, że..." albo "przepisując równanie do postaci X otrzymujemy...". Ekspert czytający taki dowód rzeczywiście od razu zauważy żądaną własność i (po włożeniu pewnego wysiłku) będzie w stanie przepisać dane równanie do właściwej postaci. Komputer zwykle sobie z tym nie poradzi. To oznacza, że matematyk chcący, aby jego dowód został automatycznie sprawdzony przez komputer musi do każdego kroku dowodu zrozumiałego dla człowieka dopisać kilkadziesiąt innych kroków zrozumiałych dla komputera - a to wymaga zbyt dużo pracy.
Obecna sytuacja jest więc następująca: z zasady wiemy jak pisać dowody matematyczne w języku logiki tak, aby mogły one być sprawdzone przez komputery. Wymaga to jednak zbyt wiele wysiłku i matematycy nie mogą poświęcić na to wystarczająco dużo czasu. Pojedyncze duże twierdzenia takie jak klasyfikacja grup skończonych czy hipoteza Keplera zostały udowodnione w ten sposób, ale kosztowało to wiele lat pracy wielu uczonych.

Niniejszy projekt zamierza wnieść wkład w rozwój języków i technik ułatwiających sprawdzanie dowodów matematycznych. Najważniejszym zadaniem projektu jest rozwój systemu komputerowego zdolnego do przeprowadzania pewnych rozumowań. Chcemy, żeby w sytuacji kiedy matematyk napisze "od razu widać, że..." komputer mógł po wykonaniu pewnych obliczeń rzeczywiście zauważyć żądaną własność.

Innym aspektem, który zamierzamy zbadać jest automatyczne wykrywanie niezdefiniowanych wartości. Wszyscy nauczyliśmy się w szkole, że nie wolno dzielić przez zero. Łatwo jest rozpoznać liczbę zero, ale co zrobić w przypadku gdy dzielimy przez skomplikowane wyrażenie, np x do kwadratu minus 10x plus 26? Czy dla pewnych x to wyrażenie może być zerem? Chcemy rozwinąć język komputerowy, w którym warunki takie jak "nie wolno dzielić przez zero" mogą być sformułowane i sprawdzone w sposób automatyczny. Chcemy, żeby matematycy mogli skoncentrować się na swoich obliczeniach i tylko zobaczyć ostrzeżenia generowane przez komputer gdy przypadkowo próbują dzielić przez zero - to ma im oszczędzić czas. Dzielenie przez zero jest tu najprostszym przykładem; istnieje wiele innych, bardziej skomplikowanych i trudniejszych do sprawdzenia warunków, które chcemy wykrywać automatycznie.
Druga część projektu polega na całkowitym sprawdzeniu pewnych dowodów matematycznych. Wybraliśmy zastosowania matematyki w fizyce, dokładniej prawa ruchu bryły sztywnej. Mechanika bryły sztywnej w szkole średniej zajmuje się tylko punktami masy - jest to spore uproszczenie, bo punkt masy nie rotuje i nie ma orientacji. W rzeczywistości obiekty fizyczne mają orientację: samochód nie może poruszać się w dowolnym kierunku, tylko do przodu i do tyłu; żeby odróżnić przód od tyłu trzeba ustalić orientację samochodu.

Wybraliśmy prawa mechaniki bryły sztywnej z kilku powodów. Po pierwsze, jest to ważna teoria używana w wielu zastosowaniach.
Po drugie, w dziedzinie fizyki bardzo niewiele dowodów zostało sprawdzonych przez komputery. Po trzecie, podejrzewamy, że w niektórych podręcznikach dla inżynierów ta teoria jest używana w sposób niewłaściwy. Błędy, które spodziewamy się znaleźć, nie są rachunkowe - polegają one raczej na niesprawdzeniu warunków, w których odpowiednie równania ruchu mogą być zastosowane. Te warunki są dość podobne do wspomnianego wyżej dzielenia przez zero, tylko bardziej skomplikowane. Dokładniej, są one związane z układami współrzędnych, w których wyraża się równania ruchu. Inżynierowie używają wielu różnych układów współrzędnych i czasami je mylą. Dodając układy współrzędnych do specyfikacji warunków, i weryfikując je w naszym systemie, zamierzamy wyeliminować tego typu błędy.