

Informatyka odgrywa dziś fundamentalną rolę w funkcjonowaniu społeczeństw w praktycznie wszystkich aspektach. Istotnym elementem tej roli jest efektywne wytwarzanie oprogramowania, które powinno być możliwie niezawodne i spełniać odpowiednie wymagania. Praktyka rozwoju oprogramowania pokazuje że jest to złożony i wymagający proces, stąd wizja jego automatyzacji, choćby częściowej, pojawiła się już na wczesnych etapach historii rozwoju informatyki.

Synteza programów polega na generowaniu kodu komputerowego automatycznie lub półautomatycznie na podstawie określonych specyfikacji lub opisów. Jest to bardzo złożony obliczeniowo problem, a obecne metody nie radzą sobie z nim wystarczająco dobrze. Dzieje się tak m.in. dlatego że konstruują one programy w sposób znacząco różny od człowieka.

Standardowym sposobem wytwarzania oprogramowania przez człowieka jest wykorzystanie wcześniej zdobytej wiedzy o języku programowania. Na przykład, gdy programista ma za zadanie napisać program, który posortuje listę liczb od najmniejszej do największej, zacznie on od napisania programu realizującego bardziej podstawową funkcję (podprogram), taką jak znajdowanie najmniejszej liczby na liście. Następnie wykorzysta ten podprogram do napisania docelowego programu, który po kolei pobiera najmniejszą liczbę z listy jeszcze nie wybranych elementów i dodaje ją do wynikowej listy. Człowiek wykorzystuje w ten sposób kompozycyjny charakter reprezentacji programów oraz możliwość tworzenia i testowania podprogramów, rozwiązując tym samym mniejsze problemy w celu dojścia do rozwiązania większego problemu.

Taki sposób wytwarzania oprogramowania wydaje się nam naturalny, natomiast algorytmom syntezy programów daleko jeszcze do takiego działania. W tym projekcie przeanalizujemy algorytmy głębokiego uczenia, stanowiącego najbardziej prężnie rozwijającą się obecnie gałąź sztucznej inteligencji i uczenia maszynowego, dla syntezy programów działające w taki iteracyjny sposób. Zbadamy, jakie sposoby reprezentowania programów dla tego typu algorytmów są najbardziej użyteczne. Dodatkowo zweryfikujemy, jakie modele głębokiego uczenia oraz algorytmy uczące najlepiej nadają się do zadania syntezy programów.

Badania przeprowadzone dotąd w naszym zespole wskazują, że takie iteracyjne podejście do procesu syntezy, sformułowane jako powiązane formułowanie hipotez i ich walidacja, pozwala systemowi uczącemu się na lepsze zrozumienie problemu oraz odkrywanie zależności pomiędzy poszczególnymi elementami programów.

Projekt ten ma na celu zapewnienie lepszego zrozumienia reprezentacji, struktur i algorytmów związanych z zastosowaniem głębokiego uczenia do problemu syntezy programów. Wyniki projektu mogą znacząco wpłynąć na dziedzinę syntezy programów i rozszerzyć się na inne dyscypliny stojące przed podobnymi wyzwaniami, a także potencjalnie wpłynąć na praktykę uczenia maszynowego, tj. wykorzystanie syntezy programów jako formy uczenia maszynowego.