

# Synteza poprzez aktywne uczenie się: teoria i zastosowania

## Streszczenie popularnonaukowe

Każdego roku produkuje się ogromne ilości kodu komputerowego, więc ułatwianie programowania i zwiększenie jego niezawodności są intensywnie badane. Ostatnio opracowano wiele narzędzi, takich jak ChatGPT, Copilot, Claude i inne, które są oparte na dużych modelach językowych (LLM). Narzędzia te w szczególności potrafią generować kod komputerowy z opisów w języku naturalnym, co ułatwia programowanie. Choć wyniki uzyskiwane przez te narzędzia są imponujące, LLM są krytykowane za częste nieścisłości w generowanym kodzie. Koncepcja automatycznego generowania kodu komputerowego nie jest jednak nowa.

W 1957 roku Alonzo Church zaproponował proces zwany *syntezą*, w którym obwody cyfrowe są automatycznie generowane na podstawie specyfikacji podanej za pomocą formuły logicznej. Choć może się to wydawać powierzchownie podobne do generowania kodu przez LLM, podstawowa idea jest zupełnie inna, ponieważ głównym celem syntezy jest niezawodność. LLM generują kod na podstawie opisów w języku naturalnym, który jest nieprecyzyjny, ale łatwy w użyciu. Z kolei systemy będące wynikiem syntezy są wierne specyfikacji do tego stopnia, że można nawet matematycznie udowodnić ich poprawność. Wiąże się to jednak z wysokimi kosztami obliczeniowymi i wymaga znacznie więcej wysiłku, aby jednoznacznie opisać system. Pomimo tych wad, synteza jest intensywnie badana od ponad 60 lat, ponieważ w niektórych obszarach, takich jak programy współbieżne lub projektowanie sprzętu, osiągnięcie poprawności jest kluczowe i trudne.

Podejścia do syntezy różnią się w zależności od dokładności specyfikacji. Obejmują one generowanie kompletnych programów na podstawie szczegółowej specyfikacji, poprzez uzupełnianie częściowo napisanych programów aby spełniały więzy (prostą specyfikację), aż po generowanie programów na podstawie przykładów bez specyfikacji, zwane „*programming by example*” (PBE). Podczas gdy podejście syntezy jedynie ze specyfikacji zainspirowało wiele badań teoretycznych, podejście PBE odniosło sukces w praktyce. Przykładowo, w programie Microsoft Excel zaimplementowano funkcjonalność FlashFill, która automatycznie generuje makra na podstawie danych (wystarczy podświetlić tabelę z pustymi komórkami i nacisnąć klawisz F5).

Specyfikacja za pomocą przykładów jest przykładem *syntezy indukcyjnej*, która opiera się na *aktywnym uczeniu się automatów*. Aktywne uczenie się automatów to technika generowania *automatów skończonych*, które są abstrakcyjnym modelem obwodów cyfrowych, za pomocą algorytmu który *aktywnie* zadaje zapytania i generuje automat na podstawie odpowiedzi. Technika ta była jednak stosowana tylko w przypadkach, w których na zapytania odpowiadano automatycznie, a nie ręcznie przez ludzi. Algorytmy zadają zbyt wiele zapytań, aby odpowiedzieć na nie ręcznie.

W tym projekcie badawczym planujemy rozwinąć teorię syntezy ze szczególnym uwzględnieniem aktywnego uczenia się automatów i zastosować wyniki teoretyczne do wspomaganej syntezy obwodów. Planujemy zrealizować następujące zadania:

**Zadanie 1: Złożoność syntezy.** Planujemy badać problemy teoretyczne motywowane syntezą. Uzyskane wyniki ukierunkują nasze badania w kolejnych zadaniach, wskazując obiecujące kierunki i te, które prawdopodobnie nie przyniosą sukcesu.

**Zadanie 2: Aktywne uczenie się z więzami.** Istnieją dwa podejścia do syntezy: synteza ze specyfikacji i synteza z przykładów. Planujemy połączyć te dwa podejścia i opracować wariant syntezy, w której algorytm syntezy przyjmuje częściową specyfikację (więzy) i przeprowadza aktywne uczenie się automatów uwzględniając więzy. Oczekujemy, że połączenie dwóch podejść znacznie zmniejszy liczbę zapytań zadawanych przez algorytmy uczące się, co sprawi, że aktywne uczenie się automatów będzie można zastosować w narzędziach interaktywnych.

**Zadanie 3: Wytłumaczalność syntezy.** Podczas gdy proces syntezy jest niezawodny, mogą wystąpić błędy w specyfikacji lub w odpowiedziach na zapytania. Aby wykryć problemy ze specyfikacją, planujemy opracować *wytłumaczalne* algorytmy syntezy oparte na aktywnym uczeniu się, które będą generowały systemy wraz z informacjami o kluczowych decyzjach podjętych w procesie syntezy.

**Zadanie 4: Wspomaganie specyfikacji układów cyfrowych.** Systemy elektroniczne są specyfikowane przy użyciu języków opisu sprzętu (HDL), takich jak Verilog lub VHDL. Planujemy opracować narzędzie ułatwiające pisanie kodu HDL poprzez syntezę części kodu przy użyciu aktywnego uczenia się automatów. W przeciwieństwie do innych podejść, naszym celem jest generowanie tylko wybranych części w programach HDL, a nie kompletnych systemów.