

Wydajne algorytmy dla słabych form niedeterminizmu

MICHAŁ SKRZYPCZAK

Automaty skończone stanowią matematyczny model maszyn wykonujących pewne działania. Typowym ich zastosowaniem jest dopasowanie tekstu do zadanego wzorca (np. wyrażenia regularnego) — automat wczytuje kolejne znaki tekstu i po jego wczytaniu stwierdza, czy tekst zgadza się ze wzorcem. W ten sposób automat może stwierdzić, czy *akceptuje* dane wejście. Wyróżnia się dwa rodzaje automatów: deterministyczne, w których kolejne akcje podejmowane przez maszynę wynikają jednoznacznie z jej aktualnej konfiguracji; oraz niedeterministyczne, gdzie dostępnych jest więcej niż jedna akcja. Aby automat niedeterministyczny *zaakceptował* dane wejście wystarczy, by wśród wielu dostępnych dla niego przebiegów na tym wejściu, chociaż jeden prowadził do akceptacji. Mówiąc żargonem, automat niedeterministyczny *zgaduje*, którą z możliwych akcji wybiera, mając za cel zaakceptowanie danego mu wejścia.

Jeden z kluczowych wyników teorii automatów mówi, że każdy automat niedeterministyczny daje się zeterminizować: można stworzyć inny automat, w pełni deterministyczny, który akceptuje dokładnie te same wejścia co oryginalny automat niedeterministyczny. Oznacza to, że daje się symulować owo *zgadywanie* przez deterministyczny wybór. Niestety odbywa się to kosztem rozmiaru automatu: istnieją automaty niedeterministyczne o n konfiguracjach takie, że najmniejszy możliwy równoważny automat deterministyczny ma 2^n konfiguracji. Innymi słowy, determinizacja takiego automatu wymaga wykładniczego wzrostu liczby konfiguracji. Taka eksplozja prowadzi do dużej złożoności czasowej rozmaitych algorytmów operujących na automatach.

Niniejszy grant dotyczy badania wzrostu liczby stanów przy standardowych operacjach na automatach wykorzystujących niedeterminizm w ograniczony sposób. Najbardziej podstawową klasą takich automatów są automaty jednoznaczne: automat niedeterministyczny jest jednoznaczny, jeśli na każdym wejściu ma co najwyżej jeden przebieg prowadzący do akceptacji. Automat taki wciąż ma za zadanie *zgadywać* kolejne akcje w miarę wczytywania wejścia, jednak skądinąd wiadomo, że co najwyżej jeden wybór doprowadzi go do akceptacji.

Wiadomo, że wiele operacji i zadań dla automatów jednoznacznych daje się wykonywać bardziej wydajnie, niż w przypadku ogólnych automatów niedeterministycznych. Przykładem takiego zadania jest problem uniwersalności: sprawdzić czy dany automat akceptuje wszystkie możliwe wejścia. W przypadku ogólnych automatów niedeterministycznych problem ten wymaga czasu wykładniczego (ściślej jest PSPACE-zupełny), natomiast dla automatów jednoznacznych istnieje wielomianowy algorytm rozwiązujący ten problem. Za główny cel niniejszego grantu wybrałem operację dopełniania: dla danego automatu \mathcal{A} skonstruować automat \mathcal{B} , taki, że \mathcal{B} akceptuje jakieś wejście wtedy i tylko wtedy, gdy \mathcal{A} go nie akceptuje. Poniższa hipoteza mówi, że operację dopełnienia automatów jednoznacznych daje się wykonywać wydajnie.

Hipoteza 1 (Colcombet). *Jeśli \mathcal{A} jest automatem jednoznacznym, to istnieje automat jednoznaczny \mathcal{B} rozmiaru wielomianowego względem \mathcal{A} , taki że \mathcal{B} jest dopełnieniem \mathcal{A} .*

Hipoteza ta dotyczy jednego z fundamentalnych problemów w teorii automatów. Praktycznie wszystkie pytania dotyczące wzrostu liczby konfiguracji automatów przy podstawowych operacjach są już od dawna rozwiązane. Fakt, że pytanie dotyczące dopełniania automatów jednoznacznych jest wciąż otwarte, świadczy o jego trudności. Jednocześnie oznacza on, że jakkolwiek postęp w tym kierunku będzie miał duże znaczenie teoretyczne. Dodatkowo, wielomianowy algorytm dopełniania dla tych automatów miałby czysto praktyczne zastosowania.

Oprócz badania Hipotezy 1 oraz jej uproszczonych wersji, projekt zakłada również badanie powiązanych pytań dla innych słabych form niedeterminizmu. Pierwsze z tych pytań to znowu pytanie o dopełnianie, ale tym razem dla automatów czytających słowa z danymi. Przykładem takiego słowa może być ciąg żądań do bazy danych — każde żądanie jest jednym ze skończenie wielu typów, ale dodatkowo niesie ze sobą unikalny identyfikator użytkownika, który je wysłał. Liczba tych identyfikatorów nie jest z góry ograniczona. Podobnie jak dla wyrażeń regularnych rozpoznających wzorce w tekście, można tutaj stosować automaty. Przykładowy warunek przez nie sprawdzany, to czy w ciągu danego dnia, trzech różnych użytkowników po kolei wysłało żądania zalogowania i natychmiastowego wylogowania. Wiadomo, że niektóre niedeterministyczne automaty z danymi nie dają się dopełniać. W ramach projektu badać będę, czy daje się efektywnie dopełniać jednoznaczne automaty z danymi.