Nr rejestracyjny: 2016/21/D/ST6/01376; Kierownik projektu: dr Wojciech Karol Czerwiński

Problem separacji w teorii automatów

Wojciech Czerwiński

3 czerwca 2016

W teorii automatów podstawowym pojęciem są automaty skończone i definiowana przez nie rodzina języków regularnych. Języki regularne posiadają również inną charakteryzację, z punktu widzenia logiki są to dokładnie te języki, które dadzą się opisać formułą monadycznej logiki drugiego rzędu (MSO). Przykładowa formuła to:

$$\exists_X (\forall_{x \in X} a(x) \wedge x + 1 \notin X) \wedge (\forall_{x \notin X} b(x) \wedge x - 1 \in X),$$

która z grubsza mówi, że istnieje zbiór X pozycji, taki, że na pozycjach z X są litery a, na pozycjach spoza X są litery b oraz pozycje naprzemiennie należą do zbioru X oraz nie należą do niego. Formuła ta definiuje język $\{(ab)^n \mid n \in \mathbb{N}\}$. Jeśli ograniczy się zbiór możliwych konstrukcji dozwolonych w logice, to nie wszystkie języki regularne będą już definiowalne przy użyciu formuł. Przykładowe ograniczenia to: używanie tylko formuł logiki pierwszego rzędu, ograniczenie na poziom zagłębienia kwantyfikatorów, brak predykatu +1, pozwolenie jedynie na kwantyfikatory egzystencjalne, itd. Otrzymuje się wtedy wiele naturalnych podklas języków regularnych.

Przy badaniu siły wyrazu tych podklas naturalnym pytaniem jest *problem charakteryzacji*: dany język regularny L, czy należy on do danej rodziny języków $\mathcal{F}$? Powiemy, że język S *separuje* języki K i L jeśli $K \subseteq S$ oraz $L \cap S = \emptyset$. Innym ważnym pytaniem jest uogólnienie problemu charakteryzacji, *problem separacji*: dane dwa języki regularne K, L, czy istnieje język $S \in \mathcal{F}$, który separuje K i L? Oba problemy można rozważać też w większej ogólności, gdy badane języki niekoniecznie są regularne, ale pochodzą z pewnej rodziny $\mathcal{G}$. Wtedy problem separacji języków z $\mathcal{G}$ przez języki z $\mathcal{F}$ jest postaci: dane dwa języki $K, L \in \mathcal{G}$, czy istnieje język $S \in \mathcal{F}$, które separuje K i L?

Problem separacji jest naturalnym i ważnym pytaniem teoretycznym, przejawia się to między innymi w tym, że odnajdujemy go w wielu pozornie niezwiązanych pytaniach. Wyobraźmy sobie, że badamy pliki baz danych XML, niektóre z nich pasują do naszego schematu DTD, a inne nie. Chcemy umieć rozróżnić, czy plik pasuje do schematu, czy nie używając algorytmu, który czyta plik od góry do dołu i używa skończonej pamięci. Okazuje się, że to jest tak naprawdę problem separacji poprawnych plików XML pasujących do schematu od poprawnych plików XML niepasujących do schematu przez pewien język regularny. Łatwo zauważyć, że powyższą historię można uogólnić do dowolnego formatu plików i dowolnej ich własności, którą chcemy badać.

Inną sytuacją, w której naturalnie występuje problem separacji jest upraszczanie i liczenie przybliżeń, przykładowo w weryfikacji. Przypuśćmy, że pewne zachowania programu, który chcemy zweryfikować są pożądane, pewne są zakazane, ale niektóre są w zasadzie neutralne, nie ma z nich pożytku, ale nie wydarzy się nic złego jeśli wystąpią. Może się jednak zdarzyć, że opis zachowań pożądanych i zakazanych jest bardzo skomplikowany, trudno więc sprawdzić, czy dokładnie te zachowania występują. Można jednak ominąć problem i obliczyć pewien język zachowań, które separuje pożądane i zakazane zachowania. Będzie to pewne przybliżenie, górne ograniczenie zachowań pożądanych zawierające niektóre zachowania neutralne. Takie przybliżenie może być dużo prostsze do analizy, ale także, co ważne, dużo łatwiejsze do zrozumienia dla człowieka, niekoniecznie zajmującego się zawodowo informatyką.

W projekcie zamierzamy zająć się teoretycznymi podstawami problemu separacji. Pierwszym celem jest zbadanie problemu separacji szerokiej klasy języków sieci Petriego przez języki regularne i pokazanie, że, być może, problem ten jest rozstrzygalny, czyli można go rozwiązać pewnym algorytmem. Drugim zadaniem jest zrozumienie które problemy separacji dla regularnych języków drzew są rozstrzygalne, drzewa w sposób naturalny pojawiają się w wielu przypadkach, np. w analizie plików XML. Trzecim celem jest zbadanie złożoności obliczeniowej problemów separacji, czyli innymi słowy opracowanie możliwie szybkich algorytmów lub udowodnienie, że takich algorytmów nigdy znaleźć się nie uda.