Logika dla informatyków

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

inf.ug.edu.pl/~amb

Podsumowanie wykładów

Logika, jej tematy

- stwierdzenia deklaratywne – wyrażenia o wartości logicznej: „pada deszcz" ale nie: „czy pada deszcz?", „deszczu, deszczu przestań padać", „co za pogoda!", ...
- stwierdzenia możemy łączyć spójnikami logicznymi
- bardziej skomplikowane są stwierdzenia z kwantyfikatorami

Problemy (m.in.):

- dokładniejszy opis rzeczy zmiennych: „zawsze jest tak, że jeśli pada deszcz, to później będzie słońce"

Związki z informatyką

- obliczalność – czy w ogóle da się obliczyć/rozstrzygnąć?
- teoria typów – standard w językach programowania
- teoria modeli – weryfikacja modelowa (model checking): weryfikator modelowy SPIN, algorytmy weryfikacji modelowej dla LTL i CTL
- logiki temporalne – pozwalające wyrażać „zmienność", logiki LTL, CTL, CTL*

Dedukcja naturalna

- dane są przesłanki,
  ...stosujemy reguły dowodzenia,
  ...i dochodzimy do wniosków
- dowody jako drzewa (dokładniej, grafy skierowane acykliczne, te same przesłanki mogą być wielokrotnie użyte)
- zapis dowodów w postaci linearnej, ze wskazaniem użytych reguł
- reguły dedukcji naturalnej – dla każdego spójnika wprowadzanie i eliminacja tego spójnika

Reguły dedukcji naturalnej

	wprowadzanie	eliminacja
$\wedge$	$\dfrac{\varphi \quad \psi}{\varphi \wedge \psi}(\wedge\, i)$	$\dfrac{\varphi \wedge \psi}{\varphi}(\wedge\, e1) \quad \dfrac{\varphi \wedge \psi}{\psi}(\wedge\, e2)$
$\vee$	$\dfrac{\varphi}{\varphi \vee \psi}(\vee\, i1) \quad \dfrac{\psi}{\varphi \vee \psi}(\vee\, i2)$	$\dfrac{\varphi \vee \psi \quad \begin{array}{c}[\varphi]\\ \dots \\ \chi\end{array} \quad \begin{array}{c}[\psi]\\ \dots \\ \chi\end{array}}{\chi}(\vee\, e)$
$\rightarrow$	$\dfrac{\begin{array}{c}[\varphi]\\ \dots \\ \psi\end{array}}{\varphi \rightarrow \psi}(\rightarrow i)$	$\dfrac{\varphi \quad \varphi \rightarrow \psi}{\psi}(\rightarrow e)$
$\neg$	$\dfrac{\begin{array}{c}[\varphi]\\ \dots \\ \bot\end{array}}{\neg\varphi}(\neg i)$	$\dfrac{\varphi \quad \neg\varphi}{\bot}(\neg e)$
$\bot$	brak	$\dfrac{\bot}{\varphi}(\bot\, e)$

Logika klasyczna vs intuicjonistyczna

Dowód niewprost:

$$\dfrac{\begin{array}{c}[\neg\varphi]\\ \dots \\ \bot\end{array}}{\varphi}(\text{niewprost})$$

1	$\neg\varphi \rightarrow \bot$	przesłanka
2	$\neg\varphi$	założenie
3	$\bot$	$\rightarrow$e 1,2
4	$\neg\neg\varphi$	$\neg$i 2-3
5	φ	$\neg\neg$e 4

- jeśli zaczniemy od założenia niewprost $\neg\varphi$, to wprowadzenie negacji da podwójne zaprzeczenie $\neg\neg\varphi$
- żeby otrzymać tezę φ trzeba skorzystać z eliminacji podwójnego zaprzeczenia

Zasada wyłączonego środka:

- $\dfrac{}{\varphi \vee \neg\varphi}(TND)$ tertium not datur

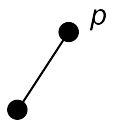
Rachunek predykatów

- zmienne, termy dotyczące dziedziny zastosowań, relacje pomiędzy termami
- kwantyfikatory, ogólny (dla każdego) i egzystencjalny (istnieje)
- problemy składniowe związane ze zmiennymi, wolne wystąpienia, wiązanie, podstawienia
- problemy nie tylko w logice
- reguły wprowadzanie i eliminacji dla kwantyfikatorów
- problem pustej dziedziny
- wprowadzanie kwantyfikatora egzystencjalnego zakłada niejawnie, że term t coś oznacza

$$\dfrac{\varphi[t/x]}{\exists\, x\varphi}(\exists\, i)$$

Semantyka rachunku zdań

- klasycznego rachunku zdań – wartościowania zmiennych zdaniowych
- zestaw reguł rachunku zdań jest poprawny i pełny
- jedynym problemem sprawdzania spełnialności może być złożoność
- semantyka intuicjonistycznego rachunku zdań – struktury Kripkego
- Najważniejsza nie-tautologia: $p \vee \neg p$



- logika klasyczna: $p \vee \neg p$ jest tautologią, choć ani p ani $\neg p$ z osobna nie muszą być prawdą
- logika intuicjonistyczna: i właśnie dlatego $p \vee \neg p$ nie jest tautologią

Semantyka rachunku predykatów

- modele – zbiory z wartościowaniem zmiennych

Twierdzenie (K. Gödel)

Klasyczny rachunek predykatów z regułami naturalnej dedukcji jest poprawny i pełny.

- modele – zbiory z wartościowaniem zmiennych

Obliczalność/ rozstrzygalność

- model obliczeń – np. maszyny Turinga
- teza Churcha – wszystkie pojęcia obliczalności są równoważne
- *input* jest parą (p, d) – dowolny program i jego dane, chcemy sprawdzać, czy ten program na tych danych się zatrzyma

Twierdzenie

Problem stopu jest nierozstrzygalny.

- dowód – ta sama sztuczka co w paradoksie Russela i w paradoksie z golibrodą

Rozstrzygalność

Twierdzenie

- *klasyczny rachunek zdań jest rozstrzygalny*
- *intuicjonistyczny rachunek zdań jest rozstrzygalny*

Twierdzenie

Logika predykatów nie jest rozstrzygalna

Twierdzenie (K. Gödel)

dla każdego modelu $\mathcal{M}$ wystarczająco bogatego (np. umożliwiającego arytmetykę) i każdego (poprawnego) systemu logicznego istnieje formuła φ prawdziwa w modelu, ale niewyprowadzalna w logice $\mathcal{M} \models \varphi$, $\mathcal{M} \not\vdash \varphi$

Algorytm badania spełnialności SAT

- problem spełnialności formuły logiki zdań
- algorytm naiwny ma złożoność wykładniczą – wszystkie wartościowania
- algorytm o niższej złożoności (w niektórych przypadkach) – etykietowanie węzłów w grafie

Twierdzenie (Cook 1971)

Problem SAT jest NP-trudny (a ściślej NP-zupełny)

Logiki temporalne i weryfikacja modelowa

Logika czasu liniowego

- model – system tranzycji (z etykietowaniem stanów)
- badamy własności prawdziwe na wszystkich ścieżkach
- operatory temporalne – następny, kiedyś, zawsze, do czasu (w kilku wersjach)
- semantyka – formuła musi zachodzic na wszystkich ścieżkach wychodzących ze stanu/stanów początkowych
- U – oznacza m.in., że warunek „do czasu" rzeczywiście zajdzie
- nie da się wyrazić możliwości, np. „możemy wrócić do początku"
- nie jest prawdą, że jeśli φ nie zachodzi, to zachodzi $\neg\varphi$ – obie formuły mogą mieć kontrprzykłady
- przykład zastosowania – problem wzajemnego wykluczenia

Logika czasu rozgałęzionego

- spójniki logiczne pozwalające kwantyfikować po ścieżkach
  - AX, AF, AG, AU
  - EX, EF, EG, EU
- semantyka – stan spełnia formułę, jeśli dla wszystkich ścieżek/ istnieje ścieżka ...w zależności od formuły
- ale formuły dla ścieżki muszą zaczynać się do X, F, G, U, nie ma możliwości powiedzieć, że np. dla wszystkich ścieżek zachodzi implikacja
- logika CTL* nie ma takich ograniczeń, osobno są formuły dla stanów na ścieżce, osobno dla ścieżek wychodzących ze stanu
- CTL* obejmuje obie logiki, LTL i CTL są nieporównywalne

Weryfikacja modelowa

- algorytm sprawdzający formułę logiki CTL – w miarę oczywiste etykietowanie stanów podformułami prawdziwymi w danym stanie
- algorytm sprawdzający formułę logiki LTL – budowa automatu falsyfikującego formułę i sprawdzenie, czy istnieje przebieg wspólny dla badanego automatu i tego falsyfikującego
- narzędzia – system SPIN

Alloy

Alloy

- system wspomagania modelowania software'u, deklaratywny – użytkownik opisuje własności
- specyfikacja – zbiór formuł, które określają zakładane okoliczności użycia software'u
- model – zbiór z działaniami spełniający specyfikację
- podstawy matematyczne – zbiory, relacje, funkcje, złączenia relacji, nieodróżnianie zbioru jednoelementowego od tego elementu
- jeśli coś jest źle, mały kontrprzykład to pokaże, a więc należy wyczerpująco zbadać zbiór małych modeli
- logika temporalna LTL możliwa do użycia stanów zmieniających się

Przykłady użycia systemu

- specyfikacja systemu plików
- system sterowania pociągami
- system zarządzania tekstem
- zagadki logiczne
- możliwości użycia logiki temporalmnej LTL w specyfikacji: system zarządzania plikami na stronie WWW, zagadki z ruchem