

# Computability Logic

Wojciech Kołowski

12 czerwca 2018

1 Motywacja

2 Droga do celu

3 Gry

# Ostrzeżenie

- Niestety zabrakło mi czasu, żeby dokończyć prezentację. Odsyłam w związku z tym do poniższej publikacji:
- A Survey of Computability Logic (gdyby komuś nie działał link, to dostępna jest pod adresem <https://arxiv.org/pdf/1612.04513.pdf>)

# Słabe punkty logik

- Logika klasyczna: każda maszyna Turinga kończy pracę lub nie... ale czy da się napisać program (albo zbudować maszynę), która będzie to rozstrzygać? Potrzeba logiki intuicjonistycznej.
- Logika intuicjonistyczna: jeżeli mam 10 zł, to mogę kupić kebab. Wobec tego za 10 zł mogę kupić dowolną ilość kebabów. Potrzeba logiki liniowej.
- Logika liniowa: być może w Smoleńsku był zamach. Potrzeba logiki modalnej.
- Logika modalna: zawsze lubię placki, więc jutro lubię placki. Potrzeba logiki temporalnej.
- Logika temporalna nie jest bynajmniej logicznym zbawicielem.

# Idee stojące za logikami

Jak widać, za każdą logiką stoi jakaś idea, do której ta logika się ogranicza, odrzucając (przynajmniej częściowo) inne.

- Logika klasyczna — prawda.
- Logika intuicjonistyczna — obliczenia.
- Logika liniowa — zasoby.
- Logiki modalne — modalności.
- Logiki temporalne — czasy.

# Być jak Sauron

- Rodzi się naturalne pytanie: czy da się stworzyć Jedyny Pierścień, który będzie rządził innymi?
- W sumie moglibyśmy zsumować wszystkie wymienione wyżej logiki i gitara, c'nie?
- Nie do końca, bo musimy jeszcze ustalić, jakie są zależności między różnymi spójnikami, np. liniową implikacją i modalnościami.
- Jeżeli pozbędziemy się idei modalności i czasu, to pewien Gruzin nazwiskiem Japaridze twierdzi, że udało mu się zbudować Jedyny Pierścień.

# Semantyka ponad składnią

- Skoro naiwne podejście składniowe (“wrzucmy do jednego worka spójniki z różnych logik i zobaczymy, co wyjdzie”) nie zadziała, to pozostaje nam podejście semantyczne.
- Należy zaznaczyć, że Japaridze uważając semantykę za ważniejszą od składni popełnia błąd — semantyka i składnia są równie istotne.

# Retoryka i prawda

- Logikę można postrzegać jako pewną grę: oto dwóch retorów przerzuca się argumentami, próbując dowieść swoich racji.
- Jeżeli przyjmiemy, że dzielają oni pewne standardy dotyczące tego, jakie argumenty są przekonujące, to możemy powiedzieć, że zdanie jest prawdziwe, gdy w odpowiadającej mu grze jeden z retorów zawsze jest w stanie przekonać drugiego do swojego zdania niezależnie od tego, jakich ten użyje kontrargumentów.



# Obliczenia interaktywne

- Obliczenia klasycznie rozumiane dotyczą funkcji, tj. maszyna dostaje coś na wejściu i ma zwrócić na wyjściu odpowiedź.
- Nie odpowiada to wielu sytuacjom spotykanym w codziennym życiu, np. dialog użytkownika z serwerem.
- Można wprawdzie modelować takie sytuacje za pomocą funkcji, ale daje to marne efekty. Np. serwer nie staje się wolniejszy wraz z każdym zapytaniem, co znaczy, że jego odpowiedzi nie są wynikami funkcji zależącej od historii interakcji z użytkownikiem.
- Jest tak dlatego, że obliczanie funkcji charakteryzuje się niskim poziomem interaktywności — interakcja sprowadza się tu do jednego zapytania i jednej odpowiedzi.

## Obliczenia interaktywne 2

- Obliczenia interaktywne również można postrzegać jako pewien rodzaj gry.
- Jeden z graczy podaje na wejściu swoje zapytanie, a drugi oblicza odpowiedź. Każdy z nich może korzystać z poprzednich zapytań i odpowiedzi.
- Możemy powiedzieć, że problem jest obliczalny, jeżeli w odpowiadającej mu grze gracz liczący ma strategię, która zawsze zwraca poprawne rozwiązanie problemu, niezależnie od tego, jakie dane drugi gracz poda na wejściu.
- Zauważmy, że zdania logiki klasycznej są specjalnym przypadkiem problemów obliczeniowych o zerowym stopniu interakcji — są prawdą lub fałszem i nie trzeba tu nic liczyć.

# Game semantics

- Odpowiednią semantyką dla naszej logiki są zatem gry.
- Jednym z graczy w naszych grach będzie maszyna, reprezentująca wykonywanie obliczeń. Drugim będzie środowisko, którego celem będzie uprzykrzanie maszynie liczenia przez zadawanie głupich pytań i podawanie dziwnych danych.
- Aby całość nie była oszukana, nakładamy na maszynę obowiązek grania jedynie według strategii algorytmicznych. Środowisko może stosować dowolne strategie.

## Game semantics 2

- Teoria gier, jako studium kooperacji i konfliktów, bardzo przydatne w ekonomii, może wydawać się dobrym narzędziem do myślenia o świecie rzeczywistym.
- Faktycznie, gry o charakterze formalnym dobrze opisują różne pojedyncze zjawiska zachodzące w świecie.
- Japardidze myli się jednak twierdząc, że logika jest najpełniejszym, spójnym, naturalnym, adekwatnym i wygodnym narzędziem do kierowania swoimi poczynaniami w życiu.

# Dualność: problemy i zasoby

- A co z zasobami?
- Nasze gry możemy interpretować jako problemy obliczeniowe, które maszyna musi rozwiązać.
- Jeżeli maszyna rozwiązuje problem, to środowisko może się jej przyglądać i wyciągać wnioski z tego procederu.
- Podobnie gdybyśmy zamienili graczy stronami i to środowisko musiało rozwiązać problem, maszyna mogłaby skorzystać z tego rozwiązania.
- Prowadzi nas to do sformułowania pewnej dualności: to co dla jednego z graczy jest problemem, dla drugiego gracza jest zasobem.

# CoL i inne logiki

- Przez skupienie się na ideach prawdy, obliczeń i zasobów CoL jest powiązana z logikami klasyczną, intuicjonistyczną i liniową.
- CoL jest konserwatywnym rozszerzeniem logiki klasycznej.
- Pewien fragment CoL jest podobny do logiki liniowej, ale występują różnice. Ogólnie zdania logiki afinicznej są słuszne w CoL, ale nie na odwrót.
- Pewien fragment CoL jest bardzo podobny do logiki intuicjonistycznej, ale występują różnice.
- Wynika to z różnego podejścia tych logik: semantycznego w przypadku CoL i syntaktycznego w przypadku logiki liniowej oraz intuicjonistycznej.

# Ekumenizm logiczny

- Logika klasyczna wszystkie wystąpienia zdania  $P$  traktuje jako ten sam zasób, zaś logika liniowa każde wystąpienie  $P$  traktuje jako osobny zasób. CoL pozwala wyrazić całe spektrum możliwości.
- O ile logika klasyczna i intuicjonistyczna pozostają w konflikcie, o tyle CoL ma charakter ekumeniczny: mamy i jedno i drugie...
- ... i jeszcze więcej: w ramach bonusu CoL jest również konserwatywnym rozszerzeniem logiki IF (*independence friendly logic*).

# Pojęcia

- Gra to drzewko, które nazywamy *gamestructure* (w wesołym tłumaczeniu: wydmuszkowa gra) i oznaczamy **Lr**.
- Wierzchołek drzewa to *run* (oznaczany  $\Gamma, \Delta$ ).
- Wierzchołek w skończonej odległości od korzenia to pozycja (oznaczana  $\Phi, \Psi, \Xi, \Omega$ ).
- Wierzchołki są utożsamiane z prowadzącymi do nich ciągami ruchów.  $\langle \rangle$  to pozycja pusta.
- Nazwa krawędzi  $(\alpha, \beta, \gamma)$  to ruch.
- Nazwa krawędzi wraz z informacją o gracz, który wykonuje ruch ( $\alpha, \beta, \gamma$  dla maszyny i  $\alpha, \beta, \gamma$  dla środowiska) to kolorowy ruch. W czarno-białym świecie oznaczane  $\top\alpha, \top\beta, \top\gamma$  dla maszyny i  $\perp\alpha, \perp\beta, \perp\gamma$  dla środowiska.



## Pojęcia 2

- *Content* to funkcja  $\mathbf{Wn} : \mathbf{Lr} \rightarrow \{\top, \perp\}$ . Mówi on, który gracz wygrywa grę w danej pozycji: jeśli  $\mathbf{Wn}(\Gamma) = \top$ , to wygrywa maszyna. W przeciwnym wypadku wygrywa środowisko.
- Gra stała (*constant game*) to para  $(\mathbf{Lr}, \mathbf{Wn})$ , gdzie pierwszy komponent to *gamestructure*, a drugi — *content*. Gry stałe uogólniają zdania znane z logiki klasycznej.
- Gra stała jest *strict*, gdy dla każdej pozycji co najwyżej jeden gracz może wykonać ruch. Gra, która nie jest *strict*, jest *free*.
- Co ciekawe, większość ciekawych gier jest *free*. Motywacja: jeżeli gramy w dwie gry na raz, to ruchy w obu grach mogą być w dowolnej kolejności.
- Strategie dla gier *free* nie mogą być funkcjami z pozycji w ruchy.

# Pojęcia 3

- Gra jest skończonej długości (*finite depth*) gdy długość gałęzi jest ograniczona przez liczbę  $d \in \mathbb{N}$ .
- Gra jest *perifinite-depth* gdy długość gałęzi jest skończona (ale może być nieograniczona).
- Gra jest skończonej szerokości (*finite breadth*), gdy ma skończoną ilość liści (pozycji końcowych).
- Gra jest skończona (*finite*), gdy ma skończoną liczbę pozycji.
- Gra elementarna (*elementary game*) to gra o głębokości 0.
- Istnieją dokładnie dwie elementarne gry stałe:  $\top$  (prawda, problem trywialny) i  $\perp$  (fałsz, problem nierozwiązywalny).

# Przykład

- Z niedostatku technologicznego przykład jest niewidzialny (gra odpowiadająca obliczaniu następnika).

# Dlaczego obliczenia interaktywne

- Wiele problemów ma więcej niż jedno rozwiązanie, np. problem “oblicz liczbę większą od  $n$ ”.
- Wiele problemów może na tym samym poziomie mieć różny kolor pozycji. Np. dla niektórych wejść funkcja częściowa jest niezdefiniowana.
- Więcej niż 2 poziomy to gry bardziej interaktywne.

## Pojęcia 4

- Gry nie muszą być stałe — mogą zależeć od zmiennych.  
Zapis:  $G(x)$ .
- Jeżeli  $G(x)$  jest grą,  $x$  zmienną, a  $c$  stałą, to  $G(c)$  jest instancją gry  $G(x)$ .
- Gra  $G(x)$  jest elementarna, jeżeli wszystkie jej instancje są elementarne.
- Gry stałe to np. szachy (bez remisów), warcaby etc.
- Gry niestałe to gry karciane — zależą one od początkowej permutacji kart.
- Pozycja jest unilegalna (*unilegal*) gdy jest legalna we wszystkich instancjach gry.
- Gra  $G(x)$  jest unistrukturalna (*unistructural*) gdy wszystkie pozycje wszystkich jej instancji są unilegalne. Intuicja: dla każdej stałej  $c$  drzewko  $G(c)$  ma taki sam kształt.

# Pojęcia 5

- Gra stała jest statyczna, gdy zwycięstwo zależy od strategii, a nie od szybkości. Jeżeli gra nie jest statyczna, to jest dynamiczna.
- Gra jest statyczna, jeżeli wszystkie jej instancje są statyczne.
- Wszystkie operacje na grach, z którymi się zapoznamy, zachowują statyczność.

# Teza (Church-Turing-Japaridze)

- Gry statyczne dobrze formalizują intuicyjne pojęcie interaktywnego (niezależnego od szybkości) problemu obliczeniowego.
- Istnienie strategii wygrywającej dobrze formalizuje intuicyjne pojęcie algorytmicznego rozwiązania interaktywnego problemu obliczeniowego.