# Sztuczna inteligencja. Logika, część II

Paweł Rychlikowski

Instytut Informatyki UWr

2 czerwca 2021

# Operacje na Bazie wiedzy

**Operacja Tell(KB, $\phi$)**

Dodaje formułę $\phi$ do bazy wiedzy (proste dodanie do zbioru)

**Operacja Ask(KB, $\phi$)**

Sprawdza, czy KB $\vdash \phi$ (czyli czy umiemy wyprowadzić $\phi$ z KB).

Czesto realizujemy operację **Ask** sprawdzając, czy $KB \land \neg\phi$ jest spełnialne/sprzeczne (dowód **nie wprost**).

**Pytanie**

Czy można użyć tu algorytmu DPLL? A WalkSat?

# Operacje na Bazie wiedzy

**Operacja Tell(KB, $\phi$)**

Dodaje formułę $\phi$ do bazy wiedzy (proste dodanie do zbioru)

**Operacja Ask(KB, $\phi$)**

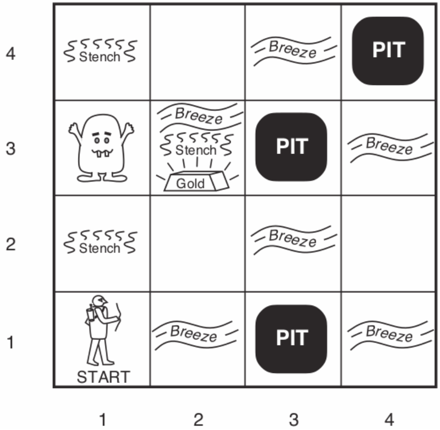
Sprawdza, czy KB $\vdash \phi$ (czyli czy umiemy wyprowadzić $\phi$ z KB).

Czesto realizujemy operację **Ask** sprawdzając, czy $KB \wedge \neg\phi$ jest spełnialne/sprzeczne (dowód **nie wprost**).

**Pytanie**

Czy można użyć tu algorytmu **DPLL**? A **WalkSat**?

# Modelowanie świata za pomocą logiki




- Wumpus śmierdzi, złoto błyszczy, w szybie są przeciągi.
- Poruszamy się o jedną kratkę w 4 kierunkach.
- Mamy jedną strzałę (strzela po liniach prostych).
- **Znamy mechanikę, ale nie znamy konkretnej edycji świata, odbieramy go za pomocą bodźców**

# Modelowanie świata za pomocą logiki zdaniowej

**Uwaga**

Musimy opisać świat za pomocą skończonej liczby bitów

Przykładowe zmienne:

1. Położenie dziur, wumpusa, złota: $P_{1,2}$, $W_{4,4}$, $G_{3,2}$
2. Położenie miejsc „z bodźcami": $S_{2,2}$, $B_{1,2}$
3. Położenie agenta: $L^t_{3,3}$ (konieczne uwzględnienie czasu)
4. Wrażenia agenta: Breeze$^t$, Stench$^t$
5. Stan agenta w chwili $t$, akcja agenta w chwili $t$, itd

# Przykładowe fragmenty modelu

- Jeżeli gdzieś jest przeciąg, to w okolicy jest dziura: $B_{1,1} \leftrightarrow P_{2,1} \vee P_{1,2}$
- Jest (co najmniej) jeden Wumpus: $W_{1,1} \vee W_{1,2} \vee \cdots \vee W_{4,4}$
- Jest co najwyżej 1 Wumpus: (w każdych dwóch W co najmniej 1 fałszywy)
- Powiązanie wrażeń agenta ze światem: $L^t_{x,y} \rightarrow (\mathrm{Breeze}^t \leftrightarrow B_{x,y})$

# Reguły ruchu

## Uwaga

Potrzebujemy dla każdej akcji agenta opisać co się zmieni, a co nie zmieni w świecie.

Przykłady:

- $L_{1,1}^{t} \wedge \text{FacingEast}^{t} \wedge \text{Forward}^{t} \rightarrow (L_{2,1}^{t+1} \wedge \neg L_{1,1}^{t+1})$
- $\text{Forward}^{t} \rightarrow (\text{HaveArrow}^{t} \leftrightarrow \text{HaveArrow}^{t+1})$
- itd

Pamiętamy, że te reguły trzeba powtórzyć dla wszystkich lokacji (i dla różnych czasów, ale o tym za chwilę)

# Hybrydowy agent w świecie Wumpusa

- Wykorzystuje procedurę szukania drogi (poruszając się po polach bezpiecznych)
- Gromadzi wiedzę o świecie:
  - Zaobserwowane bodźce (i wnioski z nich płynące)
  - „Rozwijane" zdania o mechanice świata (dla momentu $t$)
- Zarządza **planem akcji**.

# Schemat agenta hybrydowego

Mamy listę akcji do zrobienia, czyli **Plan**

Dla momentu $t$

1. Rozejrzyj się (i dodaj do Bazy Wiedzy) formuły takie jak $\mathrm{Breeze}^t$, $\mathrm{Stench}^t$, ...
2. Przeanalizuj bazę wiedzy, wyciągając możliwe konsekwencje
3. Czy trzeba zmieniać plan? Jeśli tak, to zmień (usuń wszystkie akcje, wstaw nowe)
   - **Uwaga:** musimy zmienić plan na przykład wówczas, jeżeli pierwsza akcja nie ma wypełnionych wymagań wstępnych
4. Dla akcji będącej pierwszą akcją planu wykonaj ją, czyli dodaj do bazy wiedzy formułę $\mathbf{Forward}_t$, $\mathbf{Shoot}_t$, ...

# Wumpus Agent (1)

**function** HYBRID-WUMPUS-AGENT(*percept*) **returns** an *action*
  **inputs**: *percept*, a list, [*stench*,*breeze*,*glitter*,*bump*,*scream*]
  **persistent**: *KB*, a knowledge base, initially the atemporal "wumpus physics"
    $t$, a counter, initially 0, indicating time
    *plan*, an action sequence, initially empty

  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, $t$))
  TELL the *KB* the temporal "physics" sentences for time $t$
  $safe \leftarrow \{[x, y] \ : \ \text{ASK}(KB, OK^t_{x,y}) = true\}$
  **if** $\text{ASK}(KB, Glitter^t) = true$ **then**
    $plan \leftarrow [Grab] + \text{PLAN-ROUTE}(current, \{[1,1]\}, safe) + [Climb]$
  **if** *plan* is empty **then**
    $unvisited \leftarrow \{[x, y] \ : \ \text{ASK}(KB, L^{t'}_{x,y}) = false \text{ for all } t' \leq t\}$
    $plan \leftarrow \text{PLAN-ROUTE}(current, unvisited \cap safe, safe)$

# Wumpus Agent (2)

**if** *plan* is empty and $\textsc{Ask}(KB, HaveArrow^t) = true$ **then**
  $possible\_wumpus \leftarrow \{[x, y] : \textsc{Ask}(KB, \neg W_{x,y}) = false\}$
  $plan \leftarrow \textsc{Plan-Shot}(current, possible\_wumpus, safe)$
**if** *plan* is empty **then** // no choice but to take a risk
  $not\_unsafe \leftarrow \{[x, y] : \textsc{Ask}(KB, \neg OK^t_{x,y}) = false\}$
  $plan \leftarrow \textsc{Plan-Route}(current, unvisited \cap not\_unsafe, safe)$
**if** *plan* is empty **then**
  $plan \leftarrow \textsc{Plan-Route}(current, \{[1, 1]\}, safe) + [Climb]$
$action \leftarrow \textsc{Pop}(plan)$
$\textsc{Tell}(KB, \textsc{Make-Action-Sentence}(action, t))$
$t \leftarrow t + 1$
**return** *action*

# Jeszcze o logice zdaniowej

# Dygresja. Klauzule Hornowskie

## Definicja

Klauzula Hornowska to taka klauzula, która ma **co najwyżej** jeden literał pozytywny.

## Przykłady

- $p_1$ (fakty)
- $\neg p_2$ (zaprzeczenia faktów)
- $\neg p_2 \vee p_3$ (czyli $p_2 \rightarrow p_3$)
- $\neg q_1 \vee \ldots \neg q_n \vee q_{n+1}$ (czyli $q_1 \wedge \cdots \wedge q_1 \rightarrow q_{n+1}$)

## Uwaga

Klauzule Hornowskie mają duże znaczenie w Programowaniu logicznym (programy w Prologu składają się z klauzul hornowskich).

# Modus ponens i rezolucja

Regułę **modus ponens**: dla dowolnych zmiennych zdaniowych $p$ i $q$

$$\frac{p,\ p \to q}{q}$$

możemy zapisać tak:

$$\frac{p,\ \neg p \vee q}{q}$$

(**Intuicja**: skracanie $p$ oraz $\neg p$)

Regułę powyższą można uogólnić tak, żeby operowała na dwóch dowolnych klauzulach, dających możliwość skrócenia.

**Uwaga**

Rezolucję da się uogólnić tak, żeby działała dla **logiki pierwszego rzędu** (z kwantyfikatorami)

# Rezolucja

## Definicja

Reguła **Rezolucji** ma postać:

$$\frac{p_1 \vee \cdots \vee p_k \vee r,\ q_1 \vee \cdots \vee q_n \vee \neg r}{p_1 \vee \cdots \vee p_k \vee q_1 \vee \cdots \vee q_n}$$

- Działa na klauzulach
- Jest zupełna (z aksjomatami postaci $a \vee \neg a \vee X$)
- Proste ćwiczenie: pokaż, że $a \vdash a \vee b$

# Koniec części I

# Braki logiki zdaniowej

Podstawowy brak: nie ma kwantyfikatorów, czyli pewne ogólne prawdy musimy wyrażać jako skończone alternatywy/koniunkcje.

**Przykłady**

- Każdy student jest pilny
- Pilni studenci zdają egzaminy, na które sa zapisani.
- Przynajmniej jedna osoba dostanie piątkę z AI

**Przykłady**

- $\forall x \mathrm{Student}(x) \rightarrow \mathrm{Pilny}(x)$
- $\forall x \forall e \mathrm{Student}(x) \wedge \mathrm{Pilny}(x) \wedge \mathrm{Zapisany}(s, e) \rightarrow Zdaje(s, e)$
- (...)

Jeżeli mówimy o skończonej liczbie obiektów, możemy traktować kwantyfikatory jako skróty dla koniunkcji ($\forall$) lub alternatywy ($\exists$)

**Definicja**

Logika, w której możemy używać kwantyfikatorów dla zmiennych pierwszego rzędu po „zwykłych" elementach.

**Przykłady** były na poprzednim slajdzie

# Kilka faktów o logice pierwszego rzędu

**Twierdzenie 1**

Logika pierwszego rzędu jest **nierozstrzygalna** (aczkolwiek istnieją pewne rozstrzygalne fragmenty)

- Nie ma nadziei na program, który będzie umiał dowieść każdego twierdzenia logiki 1-go rzędu w skończonym czasie
- Istnieją wszakże programy dowodzące twierdzenia, bazujące na różnych heurystykach, na przykład Otter, Vampire, Prover9, ...

# Kilka faktów o logice pierwszego rzędu

## Definicja

Fragment monadyczny logiki pierwszego rzędu to taki podzbiór tej logiki, w którym nie mamy funkcji (choć możemy mieć stałe), ani symboli relacyjnych o arności większej niż 1.

Przykład:

*Studenci chodzący na Sztuczną Inteligencję są niegłupi!*

$$\forall x(\mathrm{Student}(x) \land \mathrm{ChodziNaSI}(x) \to \mathit{NieGlupi}(x))$$

## Twierdzenie 2

Fragment monadyczny logiki pierwszego rzędu jest rozstrzygalny (spełnialność jest NEXPTIME-zupełna).

# Liczba zmiennych

**Twierdzenie 3**

Logika pierwszego rzędu z dwiema zmiennymi jest rozstrzygalna (spełnialność jest NEXPTIME-zupełna)

**Twierdzenie 4**

Logika pierwszego rzędu z trzema zmiennymi jest nierozstrzygalna

**Uwaga**

Różnych tego typu twierdzeń jest b. dużo. Różne formalizmy da się zredukować do logiki 1-go rzędu ograniczonej do jakiegoś konkretnego typu formuł.

# Czy komputery umieją dowodzić rzeczywiście ciekawe twierdzenia?

- W szczególności takie, z którymi ludzie mają kłopoty?
- (to nie jest oczywiste, choć można znaleźć przykłady, w dość specjalistycznych fragmentach matematyki)

Ale komputery potrafią sprawdzać dowody, asystować przy tworzeniu dowodów, sprawdzać przypadki, etc.

# O pomarańczach

Jaki jest związek poniższego obrazka z **Wielką Matematyką**

**Postulat Keplera (XVII w.)**

Trójwymiarowe kule w trójwymiarowej przestrzeni najciaśniej da się umieścić, gdy ich środki tworzą na płaszczyznach przekroju sześciokąty.

# O pomarańczach

Jaki jest związek poniższego obrazka z **Wielką Matematyką**

**Twierdzenie Halesa (Thomas Hales, 2015)**

Trójwymiarowe kule w trójwymiarowej przestrzeni najciaśniej da się umieścić, gdy ich środki tworzą na płaszczyznach przekroju sześciokąty.

- Pierwsze doniesienia o dowodzie twierdzenia są z 1998
- Ogólna idea: dowód na wyczerpanie (możliwości lub czytelnika)
- Dowód rozpatruje tysiące przypadków i uzasadnia, że to sa wszystkie alternatywy do rozpatrzenia.

Ostatecznie dowód został przepisany do języka logiki i zweryfikowany przez systemy wspomagające dowodzenie twierdzeń.

Podobna jest historia z twierdzeniem o 4 barwach (że każdą mapę da się pokolorować czterema kolorami (żeby żadne kraje o niezerowej wspólnej granicy nie miały tego samego koloru):

- Dowód: 1976
- Formalna weryfikacja: 2004

### Definicja

Logiki modalne są rozwinięciami logiki zdaniowej o operatory modalności, które wyrażają na przykład:

- Właściwości czasowe (kiedyś, zawsze, jutro)
- Możliwość bądź konieczność czegoś
- Przekonanie lub wiedza agenta o czymś

Dla logiki temporalnej przyjmujemy często następujące aksjomaty (wybór):

- $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$ ($\Box$ oznacza zawsze)
- $\Diamond\neg\phi \leftrightarrow \neg\Box\phi$ ($\Diamond$ oznacza kiedyś)
- $\bigcirc(\phi \vee \psi) \leftrightarrow \bigcirc\phi \vee \bigcirc\psi$ ($\bigcirc$ oznacza w kolejnym momencie czasu)

# Logika epistemiczna

Dla każdego agenta $a$ dodajemy modalność dotyczącą jego wiedzy, oznaczaną $K_a$

Przykładowe akjomaty i ich interpretacja:

- Jak agent zna przesłanki i regułę, to zna też wnioski:

$$K_i\varphi \wedge K_i(\varphi \implies \psi)) \implies K_i\psi$$

- Agenci znają tautologie

$$\text{jeżeli } M \models \varphi \text{ to } M \models K_i\varphi.$$

- To co wiemy, jest prawdziwe

$$K_i\varphi \implies \varphi$$

# Wiem, że nic nie wiem

- Jak coś wiem, to wiem że to wiem

$$K_i\varphi \implies K_iK_i\varphi$$

- Jak czegoś nie wiem, to wiem że tego nie wiem

$$\neg K_i\varphi \implies K_i\neg K_i\varphi$$

# Zagadka z zabłoconymi dziećmi

(niestety jest mniej zabawna i trochę łatwiejsza, więc zamiast niej będzie)

# Zagadka z rogaczami

(z góry wszystkich przepraszam za pewne aspekty tej zagadki, z którymi mocno się nie zgadzam)

1. Na wyspie mieszkają pary małżeńskie, wszyscy są mądrzy, logiczni i świadomi swojej mądrości.
2. Niestety żony czasami zdradzają swoich mężów (mężowie pewnie też, ale zagadka o tym milczy).
3. Zdradzonemu mężowi wyrastają rogi. Wszyscy je widzą, nie mówi się o nich, mąż ich nie widzi.
4. Mężowie są strasznie honorowi: mąż, który dowie, że był zdradzony, zabija swoją żonę wieczorem, wrzuca ciało do rzeki i nad ranem inni znajdują zwłoki
5. Pewnego dnia na wyspę przyjechał Kuglarz, który zebrał całą ludność na placu i powiedział: są wśród was rogacze! Wszyscy popatrzyli bez słowa po sobie, rozeszli się. Po tygodniu wypłynęły zwłoki.

**Wyjaśnij, co się stało!**

Jak ktoś nie zna zagadki, to powinien przestać oglądać film teraz. Zwłaszcza, że na następnych slajdach w zasadzie nie będzie odpowiedzi

# Wspólna wiedza i najsłynniejsza zagadka logiki epistemicznej

**Uwaga**

To że ja wiem coś, i ty wiesz, że ja wiem że coś, nie oznacza jeszcze, że ja wiem, że ty wiesz, że ja wiem coś.

- Wprowadza się specjalny operarator wiedzy powszechnej (common knowledge)
- Definiujemy wiedzę grupową:

$$E_G\varphi \Leftrightarrow \bigwedge_{i \in G} K_i\varphi$$

- Wprowadzamy notację:

$$E_G^n\varphi \text{ definiujemy jako } E_G E_G^{n-1}\varphi$$

oraz $E_G^0\varphi = \varphi$

# Operator wspólnej wiedzy

- Definiujemy operator:

$$C_G\varphi \Leftrightarrow \bigwedge_{i=0}^{\infty} E_G^i\varphi$$

- Zdanie Kuglarza nie jest zdaniem o zerowej informacji: wprowadza ono bowiem do bazy wiedzy wszystkich agentów formułę:

$$C_{mieszkancywyspy}\ \text{ktoś-ma-rogi}$$