Sztuczna inteligencja. Logika

Paweł Rychlikowski

Instytut Informatyki UWr

12 czerwca 2023

Koniunkcyjna postać normalna. Przypomnienie

Definicje

1. literał - zmienna albo $\neg$ zmienna
2. klauzula - $l_1 \vee \cdots \vee l_n$ (gdzie l_i to literał)
3. formuła w **CNF** - $c_1 \wedge \cdots \wedge c_n$, gdzie c_i jest klauzulą

Algorytm DPLL

- Algrytm Davisa–Putnama–Logemanna–Lovelanda (**DPLL**) jest algorytmem znajdującym spełniające podstawienie dla formuły CNF.
- Jest **zupełny** – tzn. zawsze kończy się z prawidłowym wynikiem, może działać długo

Uwaga

Stanowi bazę współczesnych SAT-Solverów (z jednym usprawnieniem, o którym jeszcze powiemy).

Algorytm DPLL (2)

Definicje

a) **Klauzula jednostkowa** (unit clause) – klauzula zwierająca 1 literał

b) **Czysty literał** – literał, który występuja tylko jako pozytywny, lub tylko jako negatywny (czyli z jedną polaryzacją).

$$x_1 \wedge \neg x_2 \wedge (x_3 \vee x_4 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee x_5)$$

Algorytm DPLL (2)

Jakie wnioskowanie można przeprowadzić korzystając z tych pojęć:

a) unit propagation – klazule jednostkowe można spełnić na 1 sposób (spełniając literał), wstawiając wartość logiczną do innych klauzul możemy zrobić nowe klauzule jednostkowe.

b) „Opłaca się” przypisywać czystym literałom wartość true (bo?).

Algorytm DPLL

Algorytm

Funkcja **DPLL**(Φ):

- dla każdej klazuli jednostkowej wykonaj unit propagation zmieniając Φ (do nasycenia)
- jeśli Φ zawiera pustą klauzulę zwróć false
- ustal wartości dla czystych literałów (zmieniając Φ)
- wybierz zmienną x (o nieokreślonej do tej pory wartości)
- zwróć **DPLL**($\Phi \land x$) or **DPLL**($\Phi \land \neg x$)

Oczywiście czasem wystarczy sprawdzić tylko jedną część rekurencyjnego wywołania!

O uczeniu klauzul

- Zauważmy, że jak w algorytmie DPLL osiągnęliśmy sprzeczność, to można myśleć, że „udowodniliśmy” twierdzenie (przy założeniu analizowanej formuły Φ):

$$(l_1 \wedge l_2 \wedge \ldots \wedge l_n) \rightarrow \text{Fałsz}$$

  gdzie l_i jest literałem użytym w i-tym momencie algorytmu (w propagacji lub w backtrackingu)
- Możemy zatem uznać, że udowodniliśmy twierdzenie: ze zbioru klauzul wynika $\neg l_1 \vee \cdots \vee \neg l_n$
- Trochę długa formuła (i nieużyteczna w propagacji), algorytm próbuje zatem znaleźć możliwie krótki ciąg literałów o ustalonych wartościach, który sam z siebie implikuje Fałsz (i zapamiętać go na przyszłość)

Alternatywa dla DPLL

- Wypada coś powiedzieć o drugim (również używanym) algorytmie, którym jest ...
  WalkSAT

Algorytm WalkSAT

1. Zaczynamy od losowego przypisania zmiennym wartości logicznych.
2. Jak wszystkie klauzule są spełnione (mają co najmniej 1 pozytywny literał), to **koniec**
3. Wybierz losową klauzulę, która jest niespełniona
4. Rzuć monetą (prawdopodobieństwo p):
   a) Orzeł: zmień wartość jednej zmiennej z klauzuli (teraz jest spełniona!)
   b) Reszka: zmień wartość tej zmiennej z klauzuli, która maksymalizuje: różnicę klauzul spełnionych i niespełnionych
5. Po określonej liczbie zmian można zrobić restart, ewentualnie zwrócić stałą **porażka**.

Właściwości WalkSAT

1. **PLUS:** Jak zakończy działanie z sukcesem, to formuła jest spełnialna (i znaleźliśmy podstawienie)
2. **PLUS:** Mamy pełną kontrolę nad czasem działania
3. **MINUS:** Nie możemy mówić o niespełnialności: porażka nic nie oznacza.

Wykorzystanie logiki zdaniowej

- Zadania z więzami realizowane za pomocą logiki zdaniowej
  Przykład: obrazki **logiczne**
- Algorytmy planowania
  Przeszukiwanie specjalnej przestrzeni stanów, w której dozwolone ruchy opisane są formułami

Obrazki logiczne jako CNF-SAT

Przykładowa reprezentacja

- Mamy zmienne (binarne) odpowiadające polom,
- Mamy zmienne typu: w wierszu 5 jest układ 00111001100

Co dalej?

Formuły

- W każdym wierszu (bądź kolumnie) jest jeden ze zgodnych ze specyfikacją układów (długa alternatywa)
- Formuły „tłumaczące” zmienne wierszowe na zmienne związane z polami:

$$W_{01101110} \to \neg X_0 \wedge X_1 \wedge X_2 \wedge \neg X_3 \wedge X_4 \wedge X_5 \wedge X_6$$

(to jest skrót dla n formuł typu $W_{01101110} \to L_i$)

Opisywanie akcji za pomocą formuł

Uwaga

Mamy czas, czyli $t \in 0, \ldots, T-1$. Używamy zmiennych mówiących o stanie świata w momencie t i o akcji w momencie t. Opisujemy mechanikę świata językiem logiki.

- Określamy zmienne prawdziwe w czasie 0,

$$\text{stoję-przed-sklepem}_0 \land \text{pusta-torba}_0 \land \text{pełen-portfel}_0 \land \ldots$$

- Określamy cel (czyli co chcemy uzyskać w czasie T) – czas T musimy zgadnąć, czyli inaczej mówiąc sprawdzać kolejne wartości, aż do skutku.

$$\text{stoję-przed-sklepem}_T \land \neg\text{pusta-torba}_T \land \neg\text{pełen-portfel}_T$$

Opisywanie akcji za pomocą formuł

Uwaga techniczna

Formuły zapisujemy często w bogatszym języku, zakładając, że system sam je przekształci do CNF-u.

- Warunki wstępne i końcowe poleceń

$$\text{strzelam}_t \rightarrow \text{mam-łuk}_t \wedge \text{mam-strzałę}_t \wedge \neg\text{mam-strzałę}_{t+1}$$

Opisywanie akcji za pomocą formuł (2)

- Frame axioms (świat się za bardzo nie zmienia). Przykład:

$$\text{smok-śpi}_t \wedge \text{czytam-gazetę}_t \rightarrow \text{smok-śpi}_{t+1}$$

  (to dla wszystkich niewpływających na siebie par zmienna-stanu, akcja)
- Explanatory frame axioms

$$\text{smok-śpi}_t \wedge \neg\text{smok-śpi}_{t+1} \rightarrow (\text{rzucam-granat}_t \vee \text{gram-na-puzonie}_t \vee \ldots$$

- Informacja, że w każdym czasie wykonuję akcję (i tylko jedną – łatwa do zakodowania w CNF.

Algorytm SATPLAN

- Zasadniczo właśnie go opisaliśmy
- **Powtórka:** opisujemy świat, szukamy spełnialności formuły dla kolejnych T, jak znajdziemy, wypisujemy wartościowania, z którego odczytujemy sekwencję akcji.

Sposób definiowania logiki (ogólnie)

Musimy podać 3 składniki:

1. **Składnię** – jak pisac formuły
2. **Semantykę** – co znaczą formuły, kiedy są prawdziwe
3. **Reguły wnioskowania** – jak z prawdziwych formuł wnioskować inne, również prawdziwe

Logiki mają różną siłę wyrazu i dają procedury o różnej złożoności (musimy **balansować** pomiędzy siłą wyrazu a obliczeniową trudnością logiki)

Reguły wnioskowania

Uwaga

Reguły wnioskowania dotyczą syntaktyki, nie semantyki.

Nasłynniejsza reguła wnioskowania

Reguła **modus ponens**: dla dowolnych zmiennych zdaniowych p i q

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

Można ją uogólnić do większej liczby przesłanek

$$\frac{p_1, \ldots, p_n,\ \ p_1 \wedge \cdots \wedge p_n \rightarrow p_{n+1}}{p_{n+1}}$$

Reguły wnioskowania

Ogólna postać reguły wnioskowania jest następująca:

$$\frac{f_1, \ldots, f_n}{g}$$

Algorytm wnioskowania

Wnioskowanie w przód (forward inference)

Powtarzaj, aż do momentu, gdy nie da się zmienić Bazy wiedzy:

- Wybierz $\{f_1, \ldots, f_k\} \subseteq \mathcal{KB}$
- Jeżeli istnieje reguła:

$$\frac{f_1, \ldots, f_n}{g}$$

  dodaj g do Bazy wiedzy

Definicja

Jeżeli powyższy algorytm dodaje f w którymś momencie do bazy wiedzy, wówczas piszemy $KB \vdash f$

2 proste uwagi o wnioskowaniu

1. Wnioskowanie w tył: Zaczynamy od tego, co chcemy udowodnić (od naszego celu).
2. Możemy myśleć o dowodzeniu twierdzeń jako o zadaniu przeszukiwania.
   (przestrzenią stanów są zbiory aksjomatów i dowiedzionych formuł, celem – zbiór zawierający docelowe twierdzenie)

Wynikanie

Przypomnienie

Formuła definiuje zbiór modeli $\mathcal{M}$, dla których jest ona prawdziwa. Podobnie można mówić o zbiorze modeli dla bazy wiedzy (czyli koniunkcji formuł).

Definicja

Mówimy KB $\models \phi$ wtedy i tylko wtedy, gdy każdy model KB będzie modelem ϕ ($\mathcal{M}(KB) \subseteq \mathcal{M}(\phi)$).

Wynikanie i dowodzenie

Definicja 1

Logika jest **poprawna**, jeżeli $M \vdash \phi$ implikuje $M \models \phi$

Definicja 2

Logika jest **zupełna**, jeżeli $M \models \phi$ implikuje $M \vdash \phi$

Uwaga

Poprawność jest konieczna, zupełność – porządana.

Zupełność i poprawność

- The truth, the whole truth, and nothing but the truth.

Przykład. Zupełność (?) modus ponens

Modus ponens **nie** jest zupełny

Przykład

$\mathcal{KB} = \{\text{deszcz}, \text{deszcz} \vee \text{śnieg} \rightarrow \text{mokry}\}$
Mokry jest prawdziwe, ale niedowodliwe.

Operacje na Bazie wiedzy

Operacja Tell(KB, ϕ)

Dodaje formułę ϕ do bazy wiedzy (proste dodanie do zbioru)

Operacja Ask(KB, ϕ)

Sprawdza, czy KB $\vdash \phi$ (czyli czy umiemy wyprowadzić ϕ 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?

Operacje na Bazie wiedzy

Operacja Tell(KB, ϕ)

Dodaje formułę ϕ do bazy wiedzy (proste dodanie do zbioru)

Operacja Ask(KB, ϕ)

Sprawdza, czy KB $\vdash \phi$ (czyli czy umiemy wyprowadzić ϕ 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**?

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 $a \vdash a \vee b$

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

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

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!

$$\forall x(\mathrm{Student}(x) \wedge \mathrm{ChodziNaSI}(x) \rightarrow NieGlupi(x))$$

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.

O upakowaniu kul w przestrzeni

- 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.

O upakowaniu kul w przestrzeni

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!

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:

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