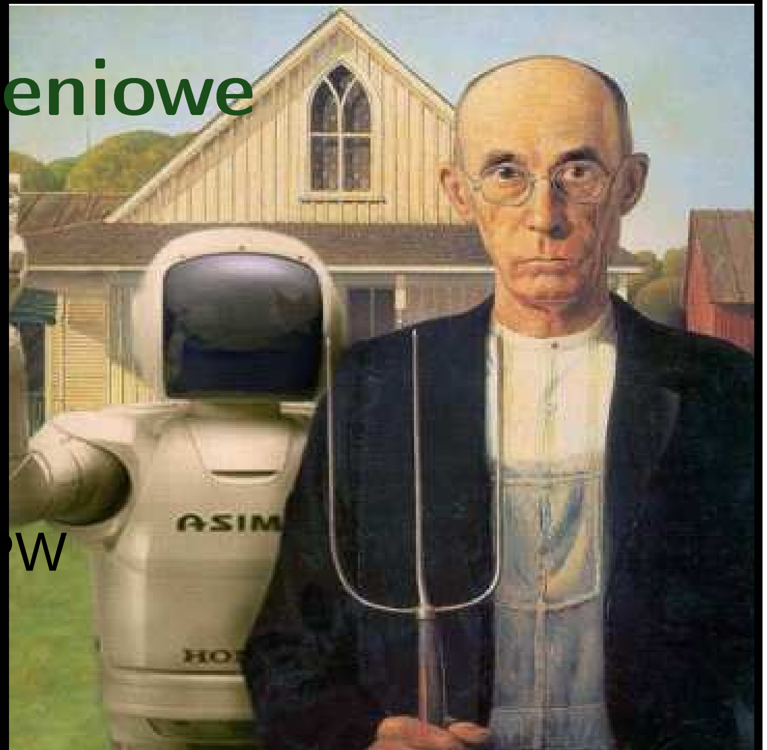
Inteligentne Systemy Obliczeniowe

Wykład 1

Piotr Wąsiewicz

Zakład Sztucznej Inteligencji - ISE PW

pwasiewi@elka.pw.edu.pl



Wprowadzenie do inteligentnych systemów

Literatura

1. M.J. Kasperski, "Sztuczna Inteligencja", Helion, 2003
2. J.J. Mulawka, "Systemy Ekspertowe", PWN, 1996
3. P. Cichosz, "Systemy uczące się", WNT, 2000
4. L. Bolc, W. Borodziewicz, M. Wójcik "Podstawy przetwarzania informacji niepewnej i niepełnej", seria Współczesna Nauka i Technika - Informatyka, PWN, 1991
5. R. Rychlik, M. Wójcik, "Od logiki do reprezentacji wiedzy", WNT - Informatyka, PWN
6. A. Skowron, "Podstawy Sztucznej Inteligencji", WNT - Informatyka, PWN
7. L. Bolc, Zaremba, "Wprowadzenie do uczenia się maszyn", PWN, 1993
8. S. Russel, P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice-Hall, 1995

Definicje Sztucznej Inteligencji (SI)

Sztuczna Inteligencja - SI - nauka o budowaniu maszyn robiących rzeczy, które, jako skonstruowane przez człowieka, wymagałyby inteligencji (M. Minsky)

Stworzenie obrazu myślącej ludzkiej istoty - stworzenie człowieka elektronicznego

Sztuczna Inteligencja - jest radykalnym wyrazem możliwości komputera cyfrowego, jest pochwałą nowej technologii

Definicje Sztucznej Inteligencji

Celem Sztucznej Inteligencji jest stworzenie systemów:

myślących jak ludzie tzn. formułujących w podobny sposób myśli np. GPS	*myślących rozumnie* tzn. formułujących myśli z pomocą komputerowych modeli np. systemy ekspertowe
działających jak ludzie tzn. o reakcjach wyglądających tak samo np. Eliza	*działających rozumnie* tzn. podających suboptymalne, satysfakcjonujące rozwiązania np. algorytmy genetyczne

Pierwsze wynalazki w dziedzinie systemów inteligentnych

Pierwsze kroki

- Arystoteles - człowiek jest zwierzęciem wyposażonym w *logos* tnz. mówienie lub pojmowanie, czy też myślenie logiczne, łaciński odpowiednik *ratio* oznacza już tylko obliczanie
- Kartezjusza *res cogitans* tzn. software (umysł) oraz *res extensa* tzn. hardware (ciało). Zwierzęta są maszynami jak mechaniczne lalki. Kartezjusza *sensus communis* to zmysł wspólny według Lema inteligencji nie da się wytworzyć w zamkniętym środowisku (mózg w słoju tylko śniłby)
- La Mettrie - umysł konsekwencją skomplikowania materii, ale sam język odróżnia człowieka od zwierząt
- W 1614 r. Jan Napier odkrywa *logarytmy* , a 8 lat później Wiliam Oughtred jest wynalazcą *suwaka* logarytmicznego, zaś w 1642 r. Błażej Pascal w konstruuje maszynę dodającą 8 cyfrowe liczby tzw. *"Pascalinę"* . Kilka lat wcześniej w 1623 r. Wilhelm Schickard, profesor z Tubingi, konstruuje podobną maszynę dodatkowo z pomocą człowieka mnożącą i dzielącą, ale idea się jeszcze nie upowszechnia

Rozwój koncepcji sztucznego umysłu

- Dalgarna "Sztuka znaków" mówi o uniwersalnym języku wszystkich ludzi np. jeśli n oznacza "żywą istotę", e - "zwierzę", k - "czworonoga", to $neke$ jest odpowiednikiem słowa "koń", $neki$ to "osioł" itd.
- Gottfried Leibniz tworzy pojęcie maszyny myślącej w sensie Leibniza systematyzuje wiedzę tworząc odpowiedni język. Ma pamięć, sensory i uczy się (dodaje nowe obiekty) oraz przeprowadza dowody w uniwersalnym języku opartym na *systemie dwójkowym* z zapisem na maszynie kulkowej przypominającej bilard (analogicznie do przepływu elektronów). W 1694 r. powstaje jego maszyna dodająca, odejmująca, mnożąca i pierwiastkująca, wcześniej niezależnie z Isaac'iem Newtonem odkrywa rachunek różniczkowy
- Charles Babbage i Ada Lovelace - maszyna licząca jak *mechaniczne* krosno "tka" wzory i nigdy nie wychodzi <u>poza</u> program. Podczas, gdy Karol myśli o konstrukcji maszyny różniczkowej, Ada marzy o maszynie grającej i malującej. Niedokończona uniwersalna maszyna "różniczkowa" Karola ma 15 ton i składa się z wielu zegarków i urządzeń napędzanych parą

Rozwój koncepcji komputerów

- Boole w połowie XIX wieku tworzy swoją algebrę operującą na liczbach w systemie dwójkowym i staje się ona podstawą najbardziej uniwersalnego języka na świecie, ale języka maszyn.
- Syn Karola, Henry Prevost, konstruuje liczydło "młynek"
- Godel ogłasza że dla dowolnego systemu formalnego, który oznaczymy jako M, zawierającego część arytmetyki liczb naturalnych jest możliwe skonstruowanie w języku systemu M takiego zdania, które nie tylko nie da się udowodnić w tym M, ale jego negacja pozostanie także bez dowodu.
- Konrad Zuse w 1936 r. opracowuje na przekaźnikach elektromagnetycznych pierwszy kalkulator Z1, a do 1941 Z2 i Z3. Wszystkie pracowały na dziurkowanej taśmie filmowej. Ostatni był Z5, do 1955 r. pracował na politechnice w Zurichu
- Polscy matematycy Marian Rejewski, Henryk Zygalski, Jerzy Różycki przekazują Anglikom prototyp "bomby" maszyny deszyfrującej komunikaty Enigmy. Przy 26! kombinacjach na każde z 26 kół szyfrujących normalne łamanie kodu trwałoby miliardy lat

Rozwój pierwszych komputerów

- Turing tworzy model teoretyczny każdego komputera tzw. *Maszynę Turinga* podobną do rybosomu organelli komórkowej czytającej nić rna (dna) i tworzącej odpowiednie łańcuchy peptydów (białka)
- Ten sam Turing w czasie wojny zajmuje się tysiącami deszyfrujących "bomb" i konstruuje potężniejsze urządzenie utrzymywane do 1976 r. w tajemnicy tzw. Collossusa składającego się z 15 tysięcy lamp i odczytującego taśmę perforowaną z prędkością 5 tys. znaków/s czyli około 50 km/h! 10 podobnych maszyn zmusiło Niemców do zmieniania szyfrów Enigmy, nie co miesiąc, ale codziennie
- John von Neumann w latach 40-stych XX wieku unowocześnia ENIACA liczącego z prędk. 5 tys. 10-cio cyfrowych liczb na sekundę, o rozmiarach 12 na 6 metrów, złożonego z 42 stalowych szaf, 19 tys. lamp, 50 tys. oporników, pobierającego 140 kWh oraz z systemem wentylacyjnym opartym na dwóch wielkich silnikach Chryslera. Przy okazji tworzy model sekwencyjnej maszyny zwanej *maszyną von Neumanna*, gdzie program i dane są przechowywane w stałej pamięci w postaci binarnej, a pobierane są sekwencyjnie do wykonania.

Współczesne idee i urządzenia

Współczesne idee machina sapiens

- Lem przytacza przykład planety gramofonu z gigantyczną pamięcią znającą odpowiedź na wszystkie pytania (np. *Deeper Blue* z programem gigantem tworzonym 6 lat, który wygrał z Kasparowem), ale z drugiej strony wspomina o opowiadaniu Dnieprowa, gdzie tranzystory zastępują ludzie i podczas procesu tłumaczenia pojedyńczy tranzystor nic nie wie o samym tłumaczeniu, gdyż przekazuje pojedyńcze litery i operacje na nich tzw. późniejszy argument *Chińskiego Pokoju Searle'a*
- Jednak mimo nieświadomości neuronów mózg ludzki ma świadomość jako całość tzw. póżniejszy argument *Jasnego Pokoju* (poruszanie magnesem w pokoju nie generuje światła, ale może jest zbyt wolne, to samo stosujemy do obliczeń i komputerowych architektur czyli magnetyzm i elektryczność jako programy wystarczą razem z energią jako składnią zdań do udowodnienia istoty światła tutaj inteligencji)

Komputerowe "mózgi" - wprowadzenie

- 1943 r. - Thomas Watson z IBM-a szacuje zapotrzebowanie na komputery na 5 sztuk, a już w latach 60-tych w każdej korporacji jest co najmniej jeden komputer np. pierwszy twardy dysk 5MB IBM-a z 1956 r. kosztował milion dolarów
- 1977 r. - Ken Olson szacuje, że nie będzie popytu na osobiste komputery, kilka lat później Steve Jobs i Stephen Wozniak sprzedają tysiące takich komputerów
- 1982 r. - człowiekiem roku tygodnika "Time" zostaje komputer
- Analogicznie możliwości powstania Internetu, czy darmowego systemu Linux nie brano w ogóle pod uwagę
- 1997 r. - Bill Gates w "Droga do przyszłości" opisuje przyszłość pełną komputerów zaspokajających codzienne potrzeby: pomoc przy leczeniu, zakupach, rozrywce. Rok później Howard Segal w "Nature" stwierdza, że to naiwne teorie, życie człowieka jest pełne uczuć, empatii, której komputery się nigdy nie nauczą
- 2005 r. - powstają japońskie wersje humanoidalnych robotów, których niedoskonałości są w większości przypadków niezauważalne

Komputerowe "mózgi" - perspektywy

- 1995 r. - w "Business Week" artykuł o zbliżającej się sztucznej inteligencji opisujący sieci neuronowe, algorytmy ewolucyjne, system ekspertowy itd.
- Inni badacze wspominają o superkomputerach o możliwościach mózgu małych ssaków, a wszyscy zapominają o znacznie większej złożoności własnych organizmów, a nawet jednej komórki, której dokładnej budowy jeszcze nie poznali, a może i nigdy nie poznają. Przykładem może być inteligencja pantofelka
- Stąd wniosek, że pojedyńczy mózg to 20 miliardów komputerów czyli komórek, których połączenia, gdyby rozplątać, złożone razem sięgałyby do Słońca (150 milionów kilometrów) lub i dalej
- Dla przykładu deseń zwykłego liścia zajmuje więcej informacji niż mieści się w 20 tomach Encyclopaedia Brytannica

Komputerowe "mózgi" - miniaturyzacja

- "There's Plenty of Room at the Bottom" Richard Feynmana - podstawowe kompendium wiedzy z 1959 roku na następne sto lat
- Główka szpilki powiększona 25000 razy umożliwia zapisanie wszystkich stron 20 tomów Encyklopedii Brittanica, gdyż najmniejsza kropka zmniejszona 25000 razy zawiera ciągle 1000 atomów
- Wszystkie książki na świecie to około 50 mln książek (np. British Museum posiada 5 mln książek) zmieszczą się na ok. 40 kartkach rozmiaru A4 lub jeśli jeden bit informacji to kostka o boku 5 atomów (125 atomów) to zmieszczą się w sześcianie wielkości najmniejszego ziarnka piasku (DNA dla przykładu zużywa 50 atomów na bit informacji)
- Miniaturyzacja ukladów scalonych powoduje przyrost mocy obliczeniowej i pojemności pamięci - po 2050 roku według prawa Moore'a (moc obliczeniowa podwaja się co 1,5 roku) komputery będą co najmniej szybsze 150 mln razy

Komputerowe "mózgi" - fakty: Deep Blue

- "Deep Blue" IBM-a 1996 r. i 1997 r. rozgrywał mecze w szachy z mistrzem świata Kasparowem
- Po 100 posunięciach jest już około 10^{150} możliwości czyli jeden ruch jest wybierany spośród wielu możliwych - przeciętnie 35 przy każdym ruchu jednego przeciwnika np. na początku 20 ruchów białych i 20 ruchów czarnych daje 400 możliwości po pierwszej kolejce
- Drzewo posunięć zawiera wszystkie partie szachowe od początku istnienia tej gry, a także wszystkie przyszłe, ale niemożliwe jest zapisanie go na jakichkolwiek dyskach choćby w odległej przyszłości. Komputer pamięta wszystkie partie arcymistrzów do 100 lat wstecz
- 200 mln operacji na sekundę w 256 procesorach czyli 50 mld w ciągu paru minut. Wraz algorytmem eliminacji słabych posunięć komputer widzi wszystkie możliwe do sytuacje 10 ruchów naprzód
- Algorytm eliminacji polega na stosowaniu heurystyk - praktycznych zasad pobranych z ludzkiego doświadczenia

Komputerowe "mózgi" - fakty: ChessBrain

- "ChessBrain" projekt podobny do SETI (search for extraterrestrial intelligence), gdzie w każdy może uruchomić program klienta, który np. w nocy będzie uczestniczył w grze w szachy lub jak w SETI przeszukiwał dane z radioteleskopów
- ChessBrain działa jak 300Ghz komputer stworzony z sieci wielu tysięcy klientów udostępniającymi swoje wolne zasoby obliczeniowe
- Najszybszy komputer ASCI Purple IBM-a (100 trylionów operacji na sekundę) zbliża się do możliwości ludzkiego umysłu. HAL z "Odysei 2001" to IBM o jedną literę do tyłu
- Kasparow w 2003 zremisował z X3D Fritz-em dowodząc tezy o złożoności ludzkiego umysłu

Futurystyczne technologie i ich perspektywy rozwoju

Komputerowe "mózgi" - przyszłość

- Mózg człowieka jest "maszyną z mięsa" dla Marvina Minsky'ego, pioniera sztucznej inteligencji
- Prof. Richard Dawkins z Oxford University uważa analogie między kodem DNA, a pamięcią komputera za początek ery postrzegania żywej materii jako niewiele różniącej się od martwej
- Za 30 lat chip tzw. "łowca dusz" zaszczepiony za okiem będzie filmował wszystkie momenty naszego życia - uważa Peter Cochrane, jeden z szefów British Telcom
- Po rewolucjach rolniczej i przemysłowej nastąpi rewolucja informacyjna - głosi futorolog Alvin Toffler czyli będziemy żyć i pracować w cyberprzestrzeni

Komputerowe "mózgi" - zabawa

- W fantastyce tzw. science fiction jednym z głównych nurtów były opowieści i nowele o sztucznej inteligencji, a w konsekwencji o robotach. Teraz coraz większą popularność zdobywają książki o cyborgach, hybrydach, nie tylko mechanicznych, ale i genetycznych np. "Impostor"
- Filmy "Terminator3", trylogia "Matrix" mówią o przewadze maszyn nad ludźmi poza kilkoma genialnymi jednostkami
- Awatary z gier komputerowych początkiem cyberprzestrzeni
- Pies AIBO czy kot NeCoRo uczą się zachowań w trakcie użytkowania, zatem są trochę inteligentne ("Blade Runner")

Idee rozwoju machina sapiens

Współczesne idee:

- Ashby stwierdza, że wszystkie układy przekazujące informację w ilości powyżej 10^8 bitów na sekundę powinny być świadome. Dzisiaj istnieją już takie komputery, ale daleko im jeszcze do inteligencji człowieka
- Adleman w laboratorium inżynierii genetycznej konstruuje komputer oparty na DNA i jego przetwarzaniu. W pojedyńczej probówce mieści się około 10^{13} odcinków DNA z zakodowaną informacją przetwarzaną masywnie równolegle
- Bariera obliczeń - problemy NP-trudne (Nondeterministic polynomial). Maszyna deterministyczna i niedeterministyczna, a równoległość i czas obliczeń. Pamięć asocjacyjna
- Problemy NP-zupełne - zadanie spełnialności zdania logicznego (funkcji logicznej)

Test Turinga: nagroda Loebnera

Warunki testu:

- Mamy co najmniej dwóch ludzi i maszynę myślącą zdolną do konwersacji
- Jeden z ludzi jest sędzią i nie widzi maszyny ani innych ludzi, z którymi porozumiewa się tylko słowami
- Jeśli sędzia po dowolnej ilości jakichkolwiek pytań, dojdzie do wniosku, że rozmawia tylko z ludźmi, to wspomniana maszyna jest maszyną myślącą w sensie Turinga
- Nagroda Loebnera jest otrzymywana za najbardziej "ludzki" komputer ang. most human computer. Test Turinga ograniczony jest do jednej dziedziny wiedzy

Wydruk dialogu z programem ELIZA

Patient: Men are all alike.

ELIZA: In what way?

Patient They're always bugging us about something or other.

ELIZA: Can you think of a specific example?

Patient: Well, my boyfriend made me come here.

ELIZA: Your boyfriend made you come here?

Patient: He says I'm depressed much of the time.

ELIZA: I am sorry you are depressed.

Maszyna myśląca

Stanowiska według Rogera Penrose'a (od "silnej AI" do "słabej AI"):

$\mathcal{A}$ Myślenie to po prostu obliczenia, a świadome doznania to wynik tych obliczeń (test Turinga wystarczający)

$\mathcal{B}$ Symulacje komputerowe świadomości nie mają nic wspólnego z samą świadomością

$\mathcal{C}$ Procesów fizycznych w mózgu nie da się zasymulować (brak dokładniejszych lub nowych praw fizyki)

$\mathcal{D}$ Świadomości nie da się wyjaśnić w żaden obliczeniowy i naukowy sposób (agnostycyzm)

Sztuczny człowiek?

Inteligentny jak człowiek program powinien

- komunikować się np. po angielsku,
- gromadzić wiedzę,
- wysnuwać na jej podstawie wnioski,
- korzystając z doświadczenia dostosowywać się do zmieniających się warunków uzupełniając wiedzę nowymi wnioskami
- oraz wykorzystywać zaawansowane systemy robotyki i wizji

Praktyczne wynalazki

Podstawowe zagadnienia SI

- Programy do prowadzenia dialogu z maszyną np. ELIZA
- Programy do rozwiązywania problemów np. GPS
- Systemy ekspertowe
- Pozyskiwanie wiedzy
- Uczenie się maszyn

Zestawienie najważniejszych osiągnięć w okresie ro

Okres	**Kluczowe osiągnięcia**
Lata przed II wojną światową	Logika formalna, psychologia poznawcza
Lata powojenne 1945-1954	Powstanie komputerów, rozwój cybernetyki
Rozpoczęcie badań w dziedzinie sztucznej inteligencji 1955-1970	Rozwój komputerów, 1956 - John McCarthy wprowadza termin "Sztuczna Inteligencja", LISP, sformułowanie programu ogólnego rozwiązywania problemów
Badania w dziedzinie rozwiązywania problemów 1961-1970	Heurystyki, robotyka, programy do gry w szachy
Systemy oparte na bazach wiedzy 1971-1980	MYCIN, HEARSAY II, MACSYMA, EMYCIN, Prolog
Po 1981 r. liczne zastosowania praktyczne	PROSPECTOR, nie zrealizowany japoński projekt komputerów piątej generacji, powstanie wielu firm zajmujących się zastosowaniem sztucznej inteligencji

Podział dziedzin sztucznej inteligencji

- rozwiązywanie problemów i strategie przeszukiwań
- teoria gier
- automatyczne dowodzenie twierdzeń
- przetwarzanie języka naturalnego (włączając przetwarzanie mowy)
- systemy ekspertowe
- robotyka
- procesy percepcji (wizja, słuch, dotyk)
- uczenie się maszyn
- wyszukiwanie informacji (inteligentne bazy danych)
- programowanie automatyczne

Zastosowania Sztucznej Inteligencji

- Deep Blue pokonał mistrza świata Gary Kasparova w 1997 r.
- Program PEGASUS rezerwuje miejsca w amerykańskich liniach lotniczych słuchając poleceń klientów
- Program ALVINN może w każdych warunkach atmosferycznych kierować ciężarówką np. przejechał nią z Washingtonu do San Diego
- Inteligentne programy rozpoznają twarze np. w bankach, odręczne pismo, sprawdzają lub projektują układy elektroniczne np. EURISKO, rekonstruują projekty architektów, szuka złóż geologicznych np. PROSPECTOR, DIPMETER, interpretuje związki chemiczne np. SCANMAT, DENDRAL
- Programy zwane systemami ekspertowymi pomagają lub są lepsze w diagnozach lekarskich np. MYCIN, CADUCEUS, CASNET, Intellipath, Pathfinder; konfigurują sprzęt komputerowy np. XCON; pomagają w podejmowaniu finansowych decyzji znajdując zdefraudowane, nietypowe lub błędne transakcje np. AMEX credit
- Programy mogą udowadniać matematyczne twierdzenia, tłumaczyć na języki obce np. Altavista, planować procesy produkcyjne, operacje w trudnych warunkach np. DART

O wiedzy

Pojęcia podstawowe

Symbol - encja reprezentująca element ze zbioru znaczeń zdefiniowanych a priori

Dane - zapisany zbiór symboli

Informacja - dane z przypisanym znaczeniem

Pojęcie - zbiór encji z jakiegoś powodu zunifikowany

Język - zbiór pojęć i reguł do tworzenia opisu rzeczywistości

Opis - wyrażenie w pewnym języku charakteryzujące obiekt lub zbiór obiektów

Wiedza - zorganizowana, uogólniona i/lub abstrakcyjna informacja

Definicje wiedzy

Wiedza deskrypcyjna to - *opisy* obiektów, ich klasy

Wiedza preskrypcyjna to - *procedury* opisujące dopuszczalne operacje, jakie można dokonać na relacjach, funkcjach tzw. przepisy

Wiedza to zbiór *faktów, reguł, domniemań* (ang. believes - fakty i reguły nie w pełni wiarygodne), *heurystyk*

Wiedza może być *prywatna* (np. inżyniera architekta), *publiczna* (ogólnodostępna), *ściśle tajna*

Wiedza może być *płytka* (opiera się na rozpoznaniu np. stylu architektury danego budynku), *głęboka* (sięga głębiej, opiera się na regułach np. dokładne poznanie wymiarów i materiałów użytych w konstrukcji budynku)

Książki to wiedza starego typu w formie *pasywnej* . Zanim zostanie ona użyta, musi być pobrana, a następnie odpowiednio zinterpretowana po czym trzeba zadecydować jak ją wykorzystać do efektywnego rozwiązywania problemu

Rodzaje wiedzy

Wiedza	Zakres	Cel(sposób)	Ważność
Pies jest ssakiem	specific	descriptive	certain
Pies ma cztery łapy	specific	descriptive	uncertain
Aby wykazać, że X jest psem należy pokazać, że rodzice X są psami	specific	prescriptive	certain
Aby udowodnić $P(X)$, wykaż, że $\neg P(X)$ jest niemożliwe ($\neg\forall x\ P(X) \Leftrightarrow \exists x\ \neg P(X)$)	general	prescriptive	uncertain
Rzeczy - obiekty rzeczywiste są obserwowalne	general	descriptive	uncertain (nie oznacza to, że zawsze je widać)

Reprezentacje wiedzy

Reprezentacja proceduralna - polegająca na określeniu zbioru procedur, których działanie reprezentuje wiedzę o dziedzinie np. $V = \frac{4}{3}\pi r^3$

Reprezentacja deklaratywna - polegająca na określeniu zbioru specyficznych dla rozpatrywanej dziedziny faktów, stwierdzeń i reguł (np. katalog rzeczy)

Zaletą reprezentacji proceduralnej jest wysoka efektywność reprezentowania procesów.

Zaletą reprezentacji deklaratywnej jest to, że jest ona bardziej "oszczędna" (każdy fakt lub reguła zapisywany tylko raz) i łatwiejsza w formalizacji.

Jako rozwiązanie optymalne można uznać reprezentację łączącą w sobie cechy reprezentacji proceduralnej i deklaratywnej np. ramy, języki obiektowe

Metody reprezentacji wiedzy

- Zastosowania logiki (rachunek zdań, rachunek predykatów, syntaktyka, semantyka)
- Zapis twierdzeń, zapis reguł w systemach ekspertowych (schemat rezolucji na klauzulach Horna, wnioskowanie w przód i wstecz)
- Wiedza nieprecyzyjna (teoria Bayesa, współczynniki niepewności w systemie MYCIN, teoria Dempstera-Shafera)
- Teoria zbiorów przybliżonych (tablice warunkowo-działaniowe, relacje nierozróżnialności, klasyfikacje, aproksymacja dolna i górna, reguły pewne i możliwe)
- Teoria zbiorów rozmytych (funkcja przynależności, liczby rozmyte, relacje rozmyte)
- Sieci semantyczne
- Algorytmy genetyczne i sieci neuronowe

Wnioskowanie

Wnioskowanie

$\mathcal{R}$ - zbiór reguł

F - zbiór faktów

Q - zbiór stosowalnych reguł

$Cond(r)$ - wszystkie warunki reguły r

$fact(r, s)$ - generowanie faktu z konkluzji reguły r poprzez podstawienia

$match(r, F, s)$ - dopasowanie faktu F do jednej z przesłanek reguły r

$match(r, g, s)$ - dopasowanie faktu g do akcji reguły r

Wnioskowanie w przód

FORWARD_CHAINING

BEGIN

  $Q := \phi$;

  REPEAT

    FOR EACH $r \in \mathcal{R}$

      IF $match(r, F, s)$ THEN

        $Q := Q \cup \{(r, s)\}$;

    IF $Q = \phi$ THEN

      BREAK;

    $(r, s) := select(Q)$;

    $Q := Q - \{(r, s)\}$;

    $F := F \cup \{fact(r, s)\}$;

  UNTIL FALSE

END

Wnioskowanie wstecz

BACKWARD_CHAINING(g)
BEGIN
    IF $g \in F$ THEN
        RETURN TRUE;
    $Q := \phi$;
    FOR EACH $r \in \mathcal{R}$
        IF $match(r, g, s)$ THEN
            $Q := Q \cup \{(r, s)\}$;
    IF $Q = \phi$ THEN RETURN FALSE;
    REPEAT
        $(r, s) := select(Q)$;
        $Q := Q - \{(r, s)\}$;
        SATISFIED:=TRUE;
        FOR EACH $c \in Cond(r)$
          IF not BACKWARD_CHAINING(c)
           THEN BEGIN; SATISFIED:=FALSE;
           BREAK; END;
        IF SATISFIED THEN
            BEGIN
                $F := F \cup \{g\}$;
                RETURN TRUE;
            END
    UNTIL $Q = \phi$;
END

Opis języka logiki

Język logiki I rzędu: syntaktyka

Wyrażenia języka składają się z term i formuł.

1. termy - encje, obiekty
   - symbole stałych (zwykle z początku alfabetu): $a, b, \ldots$
   - symbole zmiennych
   - n-argumentowe symbole funkcyjne $f(t_1, \ldots, t_n)$, gdzie $t_1, \ldots, t_n$ to termy

   np. $a, f(g(x, b), c)$ to termy zamknięty (bez zmiennych) oraz otwarty (ze zmiennymi), termy mogą być z indeksami: a_1, f_3^n, gdzie n to ilość argumentów funkcji

2. formuły - fakty zachodzące w świecie
   - formuły atomowe - symbole relacji 0-argumentowych zwanych stałymi zdaniowymi oraz relacje n-argumentowe oznaczane P tzn. $P(t_1, \ldots, t_n)$, gdzie t_n to termy dla $n \geq 1$
   - formuły - z formuł atomowych, $\neg$, $\Rightarrow$, $\forall$

     i) α, β

     ii) $(\neg\alpha)$

     iii) $(\alpha \Rightarrow \beta)$

     iv) $(\forall x \ \alpha)$, gdzie x jest zmienną

Relacja n-argumentowa oznaczana literą P jest zwana predykatem np. predykat P związany jest z pojęciem jakiejś konkretnej rzeczy tzn. $P(a)$ np. jest symbolem rzeczy osoby oznaczonej termem a np: Joanny.

Literałem pozytywnym jest α, a negatywnym $\neg\alpha$.

Język logiki I rzędu: syntaktyka

Korzystając z podanych definicji tworzenia formuł rozszerza się zbiór spójników i kwantyfikatorów języka poprzez następujące definicje:

- $(\alpha \vee \beta) = ((\neg\alpha) \Rightarrow \beta)$
- $(\alpha \wedge \beta) = (\neg((\neg\alpha) \vee (\neg\beta)))$
- $(\exists x\ \alpha) = (\neg(\forall x(\neg\alpha)))$

Symbol $\exists$ jest kwantyfikatorem szczegółowym (egzystencjalnym), $\alpha \vee \beta$ - alternatywą formuł, $\alpha \wedge \beta$ - koniunkcją formuł.

Język logiki I rzędu: syntaktyka

Formuła zamknięta zwana także zdaniem lub formułą zdaniową jest formułą bez zmiennych wolnych (zmiennych nie związanych z kwantyfikatorem $\forall$ lub $\exists$) w przeciwieństwie do formuły otwartej np. $P(x,y) \Rightarrow \exists x \forall z P(x,y)$ jest formułą otwartą. W celu poprawienia czytelności można pomijać także nawiasy kierując się następującą listą - od najmocniej do najsłabiej wiążących - spójników i kwantyfikatorów: $\neg \; \forall \; \exists \; \wedge \; \vee \; \Rightarrow$

Przykłady zdań logicznych

Piotr nie jest wysoki.

$\neg$wysoki(Piotr)

Na stole leży tylko owoc.

$\forall x$ na$(x,$ stół$) \Rightarrow$ owoc(x)

Liczba całkowita może być parzysta i nieparzysta.

$\forall x$ całkowita$(x) \Rightarrow ($parzysta$(x) \vee$ nieparzysta$(x))$

Wszyscy studenci są zdolni.

$\forall x$ student$(x) \Rightarrow$ zdolny(x)

Każdy na świecie student jest zdolny.

$\forall x$ student$(x) \wedge$ zdolny(x)

Co niektóry student jest zdolny.

$\exists x$ student$(x) \wedge$ zdolny(x) OK!

$\exists x$ student$(x) \Rightarrow$ zdolny(x) Zła składnia!

Przykłady formuł logicznych

Każdy delfin jest ssakiem.

$\forall x\ \text{delfin}(x) \Rightarrow \text{ssak}(x)$

Istnieje ssak, który znosi jaja.

$\exists x\ \text{ssak}(x) \wedge \text{znosi\_jaja}(x)$ OK!

$\exists x\ \text{ssak}(x) \Rightarrow \text{znosi\_jaja}(x)$ Zła składnia!

Każdy ogrodnik lubi słońce.

$\forall x\ \text{ogrodnik}(x) \Rightarrow \text{lubi}(x, \text{słońce})$

Wszystkie czerwone grzyby są trujące.

$\forall x (\text{grzyb}(x) \wedge \text{czerwony}(x)) \Rightarrow \text{trujący}(x)$

Żaden czerwony grzyb nie jest trujący.

$\neg \exists x\ \text{czerwony}(x) \wedge \text{grzyb}(x) \wedge \text{trujący}(x)$

$\forall x (\text{grzyb}(x) \wedge \text{czerwony}(x)) \Rightarrow \neg \text{trujący}(x)$

Są dokładnie dwa czerwone grzyby.

$\exists x \forall y\ \text{grzyb}(x) \wedge \text{czerwony}(x) \wedge \text{grzyb}(y) \wedge \text{czerwony}(y) \wedge \neg(x = y) \wedge \forall z (\text{grzyb}(z) \wedge \text{czerwony}(z)) \Rightarrow ((x = z) \vee (y = z))$

Przykłady zdań logicznych

Każdy lubi kogoś.

$\forall x \exists y$ lubi(x, y)

Ktoś lubi każdego.

$\exists x \forall y$ lubi(x, y)

Możesz kochać niektórych ludzi cały czas.

$\exists x \forall t(\text{osoba}(x) \wedge \text{czas}(t)) \Rightarrow \text{można\_kochać}(x, t)$

Można kochać każdego człowieka przez pewien okres czasu.

$\forall x \exists t(\text{osoba}(x) \wedge \text{czas}(t)) \Rightarrow \text{można\_kochać}(x, t)$

Istnieje student, który interesuje się co najmniej dwoma różnymi przedmiotami wykładanymi na jego wydziale.

$\exists x(\text{student}(x) \wedge \exists y \exists z(y \neq z \wedge \text{wykładany}(y, \text{wydział}(x)) \wedge) \text{wykładany}(z, \text{wydział}(x)) \wedge$ $\text{interesuje\_się}(x, y) \wedge \text{interesuje\_się}(x, z)))$

Semantyka

Semantyka logiczna zwana *teorią modeli* opisuje związki pomiędzy językiem, a fragmentem lub fragmentami "świata rzeczywistego". W logice fragmenty takie nazywane są strukturami.

Semantyka - struktura logiczna

$$S = (D, \mathbf{F}, \mathbf{R}, C),$$

gdzie $D \neq 0$ zwany jest dziedziną struktury, a jego elementy - obiektami struktury, $\mathbf{F}$ jest zbiorem funkcji $D^n \rightarrow D$, (R) jest zbiorem relacji w D^M, zaś C jest funkcją realizacji języka (interpretacją), która:

- każdemu symbolowi stałej przyporządkowuje jakiś obiekt z D,
- każdemu symbolowi funkcji n-argumentowej przyporządkowuje funkcję z $\mathbf{F}$,
- każdemu symbolowi predykatowemu przypisuje relację ze zbioru $\mathbf{R}$,
- każdej stałej zdaniowej przyporządkowuje wartości logiczne $\mathbf{0}, \mathbf{1}$, którym przypisuje się odpowiednio wartości *fałszu i prawdy* .

Przez $|P|_S$ oznacza się relację ze zbioru $\mathbf{R}$, którą funkcja C przyporządkowuje symbolowi P, tzn. $|P|_S = C(P)$.

Przykład świata klocków

KLOCKI

$D = (K_1, \ldots, K_7)$

góra$(K_2) = K_1$ dół$(K_1) = K_2$

K_1	
K_2	K_5
K_3	K_6
K_4	K_7

góra: $D \to D$ na: $D^2 \to \{0, 1\}$

dół: $D \to D$ nad: $D^2 \to \{0, 1\}$

$$S = (D, \mathcal{F}, \mathcal{R}, C)$$

$$\Rightarrow D, \mathcal{F}, \mathcal{R}$$

$$\Downarrow$$

$C(a) = K_1, \ldots$

$C(g) = K_7$

$C(q) =$ góra

$C(h) =$ dół

$C(P) =$ na

$C(Q) =$ nad

$$\Leftarrow$$

stałe: a, b, c, d, e, f, g

funkcje: q, h

symbole predykatowe: P, Q

zmienne: x, y

Semantyka - interpretacja zmiennych

Każdą funkcję ω, która symbolowi zmiennej x przyporządkowuje pewien obiekt z D nazywa się wartościowaniem zmiennych w $S = (D, \mathbf{F}, \mathbf{R}, C)$, a ich zbiór to Ω_S. Interpretację termu t w S przy wartościowaniu ω oznaczamy jako $I_\omega^S(t)$.

i) $I_\omega^S(t) = C(t)$, jeżeli t jest symbolem stałej;

ii) $I_\omega^S(t) = \omega(t)$, jeżeli t jest symbolem zmiennej;

iii) $I_\omega^S(f(t_1, \ldots, t_n)) = C(f)(I_\omega^S(t_1), \ldots, I_\omega^S(t_n))$;

Semantyka - interpretacja formuł

i) $I_\omega^S(\alpha) = C(\alpha)$, jeśli α jest stałą zdaniową;

ii) $I_\omega^S(P(t_1, \ldots, t_n)) = C(P)(I_\omega^S(t_1), \ldots, I_\omega^S(t_n))$;

iii) $I_\omega^S(\alpha) = \mathbf{1} - I_\omega^S(\beta)$, jeśli α ma postać $\neg\beta$;

iv) jeżeli $\alpha = (\beta \Rightarrow \gamma)$, to $I_\omega^S(\alpha) = \mathbf{1}$, jeśli $I_\omega^S(\beta) = \mathbf{0}$ lub $I_\omega^S(\gamma) = \mathbf{1}$, zaś $I_\omega^S(\alpha) = \mathbf{0}$ w przeciwnym razie

v) jeżeli $\alpha = \forall x \beta$, to $I_\omega^S(\alpha) = \mathbf{1}$, jeśli dla $\forall \omega'\ \omega' \in \omega[x]$ jest $I_\omega^S(\beta) = \mathbf{1}$, zaś $I_\omega^S(\alpha) = \mathbf{0}$ w przeciwnym razie, gdzie $\omega[x]$ to zbiór wartościowań dla wszystkich zmiennych, oprócz co najwyżej zmiennej x.

Spełnialność formuł

- α jest *spełniona (prawdziwa) w S dla* $\omega \in \Omega_S$ $\Leftrightarrow I_\omega^S(\alpha) = \mathbf{1}$
- α jest spełnialna, jeżeli $\exists S \exists \omega$ dla $\omega \in \Omega_S$, dla których $I_\omega^S(\alpha) = \mathbf{1}$
- *Spełnialność* formuł zdaniowych zależy jedynie od struktury
- α jest spełniona $\Rightarrow S$ jest <u>modelem (semantycznym)</u> formuły α
- Modelem zbioru formuł jest struktura, która jest modelem dla każdej formuły z tego zbioru.

Semantyczna konsekwencja

α jest *semantyczną konsekwencją* zbioru formuł $\Phi \Leftrightarrow$ dowolny model zbioru Φ jest modelem formuły α, co zapisujemy

$$\Phi \models \alpha$$

Dla $\Phi = 0$ α jest zawsze prawdziwa i zwana *tautologią*

$$\models \alpha$$

α jest *równoważna* $\beta \Leftrightarrow \alpha \models \beta$ i $\beta \models \alpha$
α jest *modelowo równoważna* $\beta \Leftrightarrow \alpha \models \beta$

Przykładowa formuła świata klocków

$$\begin{aligned}
\alpha = (\neg P \vee Q) &\quad\Leftrightarrow\quad (\neg Q \Rightarrow \neg P)\\
\alpha = (P \Rightarrow Q) &\quad\Leftrightarrow\quad (\neg Q \Rightarrow \neg P)
\end{aligned}$$

$$\begin{array}{rcl}
{}^{1)}\alpha(P=1,Q=1) & = & 1\\
(\neg 1 \vee 1) & \Leftrightarrow & (\neg 1 \Rightarrow \neg 1)\\
(0 \vee 1) & \Leftrightarrow & (0 \Rightarrow 0)\\
1 & \Leftrightarrow & 1\\
{}^{2)}\alpha(P=1,Q=0) & = & 1\\
(\neg 1 \vee 0) & \Leftrightarrow & (\neg 0 \Rightarrow \neg 1)\\
(0 \vee 0) & \Leftrightarrow & (1 \Rightarrow 0)\\
0 & \Leftrightarrow & 0
\end{array}$$

$$\begin{array}{rcl}
{}^{3)}\alpha(P=0,Q=1) & = & 1\\
(\neg 0 \vee 1) & \Leftrightarrow & (\neg 1 \Rightarrow \neg 0)\\
(1 \vee 1) & \Leftrightarrow & (0 \Rightarrow 1)\\
1 & \Leftrightarrow & 1\\
{}^{4)}\alpha(P=0,Q=0) & = & 1\\
(\neg 0 \vee 0) & \Leftrightarrow & (\neg 0 \Rightarrow \neg 0)\\
(1 \vee 0) & \Leftrightarrow & (1 \Rightarrow 1)\\
1 & \Leftrightarrow & 1
\end{array}$$

Wnioskowanie

<u>Wnioskowanie</u> to inaczej *inferencja* .

<u>Regułą wnioskowania</u> nazywamy dowolną operację, która skończonemu ciągowi formuł $\alpha_1, \ldots, \alpha_n$ nazywanych *przesłankami* (ang. premises), przyporządkowuje formułę β, nazywaną *wnioskiem* (ang. conclusion), co zapisuje się jako:

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta}$$

Reguła *modus ponens* zwana także regułą odrywania ma postać

$$\frac{\alpha, \alpha \Rightarrow \beta}{\beta}$$

Reguła *uogólniania* ma postać

$$\frac{\alpha}{\forall x \ \alpha}$$

Proces inferencji

Formuła α jest *wyprowadzalna* ze zbioru formuł Φ za pomocą reguł ze zbioru $\mathcal{R} \Leftrightarrow \exists$ ciąg formuł $\beta_1, \ldots, \beta_k$ taki, że:

i) $\alpha = \beta_k$;

ii) $\forall (i \leq k)\ \beta_i \in \Phi$ lub β_i jest wnioskiem reguły należącej do $\mathcal{R}$ z pewnych formuł z $\{\beta_1, \ldots, \beta_{k-1}\}$.

Ciąg formuł $\beta_1, \ldots, \beta_k$ nazywamy *dowodem* formuły β_k z Φ z zastosowaniem reguł wnioskowania z $\mathcal{R}$.

Teoria

Teoria jest sformalizownym opisem świata rzeczywistego i składa się z *języka* czyli zbioru formuł oraz struktury dedukcyjnej: zbioru *aksjomatów logicznych* , zbioru *aksjomatów specyficznych* i zbioru *reguł wnioskowania* .

<ins>Aksjomaty logiczne</ins> muszą być *tautologiami* .

Niech α, β, γ będą dowolnymi formułami teorii. Typowe aksjomaty logiczne teorii pierwszego rzędu to:

1. $\alpha \Rightarrow (\beta \Rightarrow \alpha)$
2. $(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$
3. $(\neg\beta \Rightarrow \neg\alpha) \Rightarrow ((\neg\beta \Rightarrow \alpha) \Rightarrow \beta)$
4. $\forall x\ \alpha(x) \Rightarrow \alpha(t)$,
   gdzie t jest termem, $\alpha(x)$ - formułą, zaś $\alpha(t)$ - formułą $\alpha(x)$ po zastąpieniu każdego wolnego wystąpienia zmiennej x termem t, ponadto $\exists$ zmienna z termu $t \Rightarrow \forall$ wolne wystąpienie zmiennej x nie leży w zasięgu działania kwantyfikatorów $\forall z$ lub $\exists z$.
5. $\forall x\ (\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \forall x\ \beta)$,

Reguły wnioskowania, a teoria

Jako reguły wnioskowania przyjmuje się regułę *modus ponens* i *uogólniania* .

Możliwy jest dobór innych aksjomatów logicznych i reguł wnioskowania np. system Gentzena.

Skoro aksjomaty logiczne oraz reguły wnioskowania są ustalone, to teorię określa się lub też w praktyce utożsamia ze zbiorem aksjomatów specyficznych.

Aksjomaty specyficzne

Aksjomaty specyficzne są formułami, które arbitralnie zostały uznane przez twórców teorii za *prawdziwe* , a opisujące cechy świata rzeczywistego np. formuła

$$(\forall x)(czowiek(x) \Rightarrow miertelny(x))$$

opisuje fakt, że każdy człowiek jest śmiertelny.

Twierdzenie

α jest *wyprowadzalna w teorii* T lub jest *jej twierdzeniem* $\Leftrightarrow$ α jest wyprowadzalna za pomocą reguł wnioskowania tej teorii z formuł pochodzących ze zbiorów jej aksjomatów logicznych i specyficznych, co zapisujemy tak

$$T \vdash \alpha$$

W przypadku rachunku predykatów (brak aksjomatów specyficznych) zapis jest następujący:

$$\vdash \alpha$$

Niesprzeczność teorii

Teoria jest *niesprzeczna* $\Leftrightarrow \forall\alpha\ \neg\alpha$ i α nie są jednocześnie twierdzeniami tej teorii. Można wykazać, że

- Teoria jest niesprzeczna $\Leftrightarrow \exists$ model tej teorii
- Dla dowolnej niesprzecznej teorii istnieje *przeliczalny* model.

α jest twierdzeniem niesprzecznej teorii $\Leftrightarrow \alpha$ jest prawdziwa w dowolnym modelu tej teorii, co formalnie można zapisać

$$T \vdash \alpha \Leftrightarrow T \models \alpha$$

Zupełność teorii

Teoria jest *zupełna* $\Leftrightarrow \forall$ zamkniętej formuły α tej teorii, albo $T \vdash \alpha$ albo $T \vdash \neg\alpha$. Taka teoria opisuje wszystkie informacje związane z reprezentowanym przez nią światem.

Rozstrzygalność teorii

Teorię nazywa się *rozstrzygalną* , jeżeli można w skończonej liczbie kroków stwierdzić, czy dowolna formuła należąca do języka tej teorii jest, czy też nie jest jej twierdzeniem.

Teorię nazywa się *półrozstrzygalną* , jeżeli można w skończonej liczbie kroków udowodnić każde twierdzenie tej teorii. Nie ma jednak gwarancji, na efektywne określenie, czy dana formuła nie jest twierdzeniem w T. Teorie I rzędu w ogólnym przypadku nie są rozstrzygalne, lecz są półrozstrzygalne. Niemniej jednak istnieją pewne rozstrzygalne klasy formuł, np: formuły z predykatami jednoargumentowymi lub poprzedzone tylko kwantyfikatorami ogólnymi lub poprzedzone tylko kwantyfikatorami egzystencjonalnymi.

Monotoniczność

Zbiór twierdzeń teorii I rzędu zwiększa się wraz ze wzrostem aksjomatów specyficznych. Własność ta nazywa się *monotonicznością* .

Standaryzacja formuł

Standaryzacja polega na przekształceniu formuł wyjściowych w formuły, które cechują się tym, że

- wszystkie kwantyfikatory wyprowadzane są na początek formuły - *postać preneksowa normalna* ;
- kwantyfikatory egzystencjalne zostają wyeliminowane - *postać normalna Skolema* $F_S \models F$;
- wyrażenie pod kwantyfikatorami jest koniunkcją alternatyw.

Z koniunkcji alternatyw przechodzi się do ich zbioru. Jeśli alternatywa jest złożona tylko z formuł atomowych pozytywnych i negatywnych, to nazywa się *klauzulą* .

Przykład standaryzacji

$$\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$$

$$\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$$

$$\neg(\neg\alpha) \equiv \alpha$$

$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$$

$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$

$$\neg\forall x\alpha \equiv \exists x\neg\alpha$$

$$\neg\exists x\alpha \equiv \forall x\neg\alpha$$

$$\alpha \wedge (\alpha \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$\alpha \vee (\alpha \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

$$Q \in \{\exists, \forall\}$$

$$Qx\alpha \vee \beta \equiv Qx(\alpha \vee \beta)$$

$$Qx\alpha \wedge \beta \equiv Qx(\alpha \wedge \beta),$$

gdzie β bez wolnych zmiennych

$$\forall x\alpha \wedge \forall x\beta \equiv \forall x(\alpha \wedge \beta)$$

$$\exists x\alpha \vee \exists x\beta \equiv \exists x(\alpha \vee \beta)$$

$$Qx\alpha \equiv Qx\alpha[x/y],$$

gdzie y bez wolnych zmiennych,

a x z wolnymi zmiennymi

CNF: $\neg p \vee \ldots \vee p$

DNF: $\neg p \wedge \ldots \wedge p$ - fałsz!

PNF: $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \mu$,

gdzie μ jest koniunkcją alternatyw

Klauzulę postaci

$$\neg\beta_1 \vee \ldots \vee \neg\beta_m \vee \gamma_1 \vee \ldots \vee \gamma_n$$

nazywa się *klauzulą Horna* $\Leftrightarrow n = 0$ lub $n = 1$ dla $m \geq 0$.

Inny zapis to

$$\beta_1 \wedge \ldots \wedge \beta_m \Rightarrow \gamma$$

Schemat rezolucji

Schematem rezolucji (Robinson - 1965) nazywa się *regułę inferencyjną*

$$\frac{A \vee B, C \vee \neg B}{A \vee C},$$

gdzie A, B, C są formułami, $A \vee C$ jest *rezolwentą binarną* klauzul wejściowych, a *klauzula pusta (NIL)* nie jest spełniona w żadnej strukturze.

Do danego zbioru klauzul Φ dołącza się zbiór klauzul modelowo równoważnych negacji formuły $\neg\alpha$, którą zamierzamy udowodnić. Potem stosuje się wielokrotnie schemat rezolucji. Uzyskanie rezolwenty równej NIL oznacza, że zbiór $\Phi \cup \{\neg\alpha\}$ jest sprzeczny i α nie jest twierdzeniem.

Przykład uczącego się robota

$K_{1,4}$	$K_{2,4}$	$K_{3,4}$	$K_{4,4}$
$K_{1,3}$ Kuch-nia	$K_{2,3}$	$K_{3,3}$	$K_{4,3}$
$K_{1,2}$ Robot $c_{1,2}$	$K_{2,2}$ $\neg c_{2,2}$	$K_{3,2}$	$K_{4,2}$
$K_{1,1}$ $\neg c_{1,1}$	$K_{2,1}$ $\neg c_{2,1}$	$K_{3,1}$	$K_{4,1}$

R_1: *modus ponens lub rezolucja*

$\neg K_{1,1} \wedge \neg K_{1,2} \wedge \neg K_{2,1}$

eliminacja $\wedge$: $\neg K_{1,1}, \neg K_{1,2}, \neg K_{2,1}$

$R_2 : \neg K_{1,1}, \neg K_{2,1}, \neg K_{2,2}, \neg K_{3,1}$

$R_3 : \neg K_{2,1}, \neg K_{1,2}, \neg K_{2,2}, \neg K_{3,2}, \neg K_{2,3}$

$R_4 : K_{1,3} \vee K_{1,2} \vee K_{2,2} \vee K_{1,1}$

rezolucja $\Rightarrow$

c - zapach kawy w pokoju (i, j)

k - kuchnia w pokoju (i, j)

Wiedza: $\neg c_{1,1}$, $\neg c_{2,1}$, $\neg c_{2,2}$, $c_{1,2}$

Korzystając ze zmysłu zapachu robot znajduje kuchnie!

$R_1 \quad \neg c_{1,1} \Rightarrow (\neg K_{1,1} \wedge \neg K_{1,2} \wedge \neg K_{2,1})$

$R_2 \quad \neg c_{2,1} \Rightarrow (\neg K_{1,1} \wedge \neg K_{2,1} \wedge \neg K_{2,2} \wedge \neg K_{3,1})$

$R_3 \quad \neg c_{2,2} \Rightarrow (\neg K_{2,1} \wedge \neg K_{1,2} \wedge \neg K_{2,2} \wedge \neg K_{3,2} \wedge \neg K_{2,3})$

$R_4 \quad c_{1,2} \Rightarrow (K_{1,2} \vee K_{2,2} \vee K_{1,1} \vee K_{1,3})$

$R_4 \vee \neg K_{1,1} : K_{1,3} \vee K_{1,2} \vee K_{2,2}$

$R_4 \vee \neg K_{1,1} \vee \neg K_{2,2} : K_{1,3} \vee K_{1,2}$

$R_4 \vee \neg K_{1,1} \vee \neg K_{2,2} \vee \neg K_{1,2} : K_{1,3}$