

Algorytmika i inteligencja

Pytanie, czy komputery potrafią myśleć, jak ktoś kiedyś powiedział, jest tak samo jak pytanie o to, czy okręty podwodne potrafią pływać. Analogia jest całkiem trafna. Chociaż wszyscy mniej więcej wiemy, do czego zdolne są łodzie podwodne – i rzeczywiście potrafią robić rzeczy podobne do pływania – „prawdziwe” pływanie jest czymś, co kojarzymy z bytami natury organicznej, takimi jak ludzie i ryby, a nie z łodziami podwodnymi. Podobnie, chociaż czytelnik musi już mieć całkiem dobre pojęcie o tym, czym jest algorytmika, a co za tym idzie możliwościami komputerów, prawdziwe myślenie kojarzy się w naszych umysłach z ludźmi, a może także z małpami i delfinami, ale nie z kolekcją bitów i bajtów oparte na krzemie. Ten nieco mniej techniczny rozdział dotyczy relacji między obliczeniami maszynowymi a ludzką inteligencją. Problem jest obciążony emocjonalnie i prawie wszystko, co o nim mówi, wywołuje gorące kontrowersje. Ze swojej strony postaramy się jak najlepiej uniknąć tych kontrowersji. Zamiast tego omówimy znaczenie lub nieistotność opisanej tutaj dziedziny algorytmiki dla komputerowej symulacji ludzkiej inteligencji. Wskazywane będą głównie problemy i trudności, a nie rozwiązania. Obszarem badań najbardziej istotnym dla tego rozdziału jest sztuczna inteligencja, w skrócie AI. W większości miejsc sztuczna inteligencja prowadzona jest na wydziałach informatyki, podczas gdy w innych istnieje albo osobny wydział sztucznej inteligencji, albo międzywydziałowa grupa związana być może z informatyką, psychologią poznawczą i neurobiologią. Obecnie na wielu uniwersytetach wszystkie te dyscypliny zostały połączone w centra nauki o mózgu lub kognitywistyki. Fakt, że jest to rozdział zamykający, nie powinien prowadzić czytelnika do wniosku, że jedynym powodem studiowania algorytmiki jest możliwość, że pewnego dnia będziemy w stanie skomputeryzować inteligencję. Jakkolwiek intrygujący może zabrzmieć ten pomysł, zadanie napisania tej książki nie zostało podjęte w celu zrelacjonowania naukowych podstaw sztucznej inteligencji. Algorytmika ma swoje własne zalety i można z łatwością przedstawić mocne argumenty za jej znaczeniem, głębią i stosowalnością, niezależnie od tego, czy rozważany jest temat tego rozdziału, czy nie. Tutaj mamy do czynienia z nowym, „miękkim” wymiarem. Fascynujący i ekscytujący, ale też kontrowersyjny i spekulacyjny. Przede wszystkim, jak zobaczymy, jest zupełnie inaczej. W kategoriach czysto technicznych można by powiedzieć, że niniejszy rozdział koncentruje się na innej swobodzie, którą można wykorzystać przy projektowaniu algorytmów, którą należy dodać do paralelizmu i rzucania monetą, o których mowa w rozdziałach 10 i 11. Ta swoboda obejmuje wprowadzenie zasady praktyczne, wykształcone domysły lub, używając przyjętego terminu, heurystyki. Charakter tej nowej placówki i jej motywujące przykłady odróżniają ją od praktycznie wszystkich dotychczas omawianych zagadnień, dlatego też potraktowano ją jako ostatnią. Podczas gdy heurystyki reprezentują szczególną naturę części sterującej inteligentnych programów, istnieją również trudności związane z przedstawianiem i manipulowaniem odpowiednimi dla nich danymi, a mianowicie wiedzą w jej różnych postaciach.

Niektóre historie robotów

Mówiąc o pływaniu, oto co wydarzyło się w latach 70., kiedy w niektórych kręgach istniało wiele naiwnych emocji wokół możliwości zbudowania naprawdę inteligentnych robotów, w jednym z najbardziej szanowanych ośrodków amerykańskiej informatyki. Delegacja Marynarki Wojennej Stanów Zjednoczonych odwiedziła centrum, aby dowiedzieć się, czy mogliby wykorzystać zgromadzoną tam wiedzę do zbudowania robota, który samodzielnie byłby w stanie nurkować pod statkami i wykonywać prace konserwacyjne w zanurzeniu. Naukowcy z dumą pokazali tym ludziom wyniki ich ostatnich wysiłków w robotyce – skomputeryzowane ramię robota połączone z kamerą wideo, które mogło czytać, rozumieć i wykonywać polecenia, takie jak „zbuduj wieżę z trzech bloków” lub „umieść czerwona piramida na niebieskim bloku.” Z grzeczności członkowie delegacji wysłuchali, obserwowali, a następnie grzecznie wyszli, głęboko rozczarowani. Ci ludzie nie mieli pojęcia o niewiarygodnych

trudnościach związanych z osiągnięciem nawet tak przyziemnych zachowań w automatycznym systemie komputerowym. Na pewnym etapie rozwoju oprogramowania sterującego ramię robota próbowało budować wieże z trzech bloków, zaczynając od góry! „Nauczenie” robota grawitacji, nawet w jego bardzo ograniczonym świecie, składającym się z kilku bloków i piramid, nie było sprawą błahą. Inna historia z udziałem tej samej grupy badaczy opowiada o firmie handlowej, która twierdziła, że wyprodukowała robota wykonującego rutynowe czynności porządkowe w odpowiedzi na polecenia wydawane prostym językiem, takie jak „zmyć naczynia” lub „przygotować obiad dla czterech osób”. Firma miała zademonstrować publicznie możliwości robota w jednym z lokalnych domów towarowych. Naukowcy z grupy, doskonale wiedząc, co można osiągnąć przy obecnym stanie wiedzy, byli pewni, że to oszustwo. Tak więc, podczas gdy wielu widzów, którzy zebrali się, aby zobaczyć cud, było zajętych oglądaniem sceny z przodu, byli zajęci za kulisami, próbując odkryć sztuczkę. Pewnie po chwili znaleźli osobę, która za pomocą nadajnika radiowego fizycznie sterowała ruchami robota, sprawiając wrażenie, że reaguje na polecenia instruktora. Historie te wydają się zaprzeczać stwierdzeniu zawartemu w Części 1, co oznacza, że komputery mogą sterować niezwykle wyrafinowanymi robotami przemysłowymi, które konstruują złożone elementy składające się z wielu komponentów. Nie ma sprzeczności. Roboty te są zaprogramowane do wykonywania długich i skomplikowanych sekwencji operacji według starannie przygotowanej receptury. Ogólnie rzecz biorąc, można je przeprogramować do wykonywania różnych zadań, a czasami są w stanie dostosować swoje zachowanie, aby dostosować się do zmieniających się sytuacji. Nie są jednak w stanie spojrzeć na swoje otoczenie, zdecydować, co należy zrobić, a następnie ułożyć plan i wykonać go do końca. Z tego powodu nikt nie wie, jak zaprogramować robota do budowy ptasiego gniazda ze sterty gałązek. Odnotowano pewne sukcesy w radzeniu sobie z bardzo ograniczonymi światami bloków i piramid, ale nie z gałązkami o różnych kształtach i rozmiarach lub z dużą i różnorodną gamą elementów maszyn. Radzenie sobie z nimi wymaga poziomów inteligencji, które znacznie przekraczają dzisiejsze możliwości. Nawet umiejętność uchwycenia prostej sceny, takiej jak normalny salon (przy użyciu jakiegoś wizualnego sprzętu sensorycznego) i „zrozumienia” jej, coś, co każde dziecko może zrobić, wykracza daleko poza obecne możliwości. Inny przykład podany w rozdziale 1 dotyczył kontrastu między tomografią komputerową (syntetyzującą przekrój mózgu z dużej liczby zdjęć rentgenowskich wykonanych pod różnymi kątami) a możliwością wywnioskowania wieku osoby ze zwykłego zdjęcia. Tutaj też nie ma sprzeczności. Podczas gdy pierwsze zadanie realizowane jest za pomocą skomplikowanych, ale dobrze zdefiniowanych procedur algorytmicznych, drugie wymaga prawdziwej inteligencji. Komputeryzacja inteligencji, uczynienie jej algorytmiczną, to coś, o czym wiemy zdecydowanie za mało.

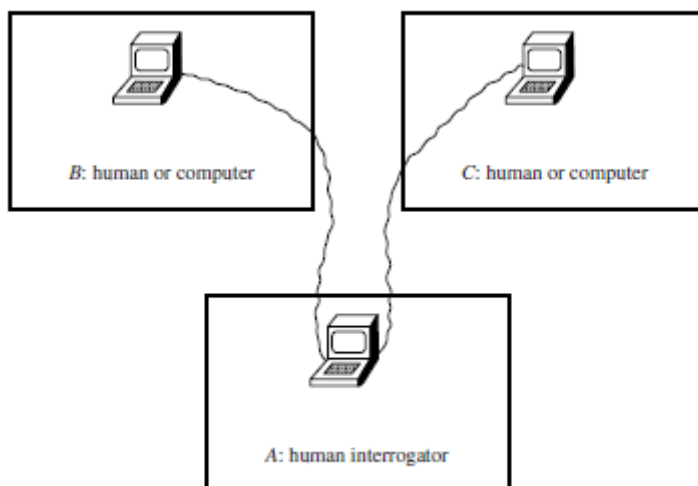
Inteligencja algorytmiczna?

Czym jest inteligencja? Konkurując z filozofami, psychologami poznawczymi i badaczami sztucznej inteligencji w tej kwestii, nic nie można zyskać. Jednak z punktu widzenia laika wydaje się, że samo pojęcie sztucznej inteligencji, czy też, żeby zmienić nazwę, by pasowało do reszty książki, inteligencja algorytmiczna, jest sprzecznością w kategoriach. Mamy tendencję do postrzegania inteligencji jako naszej podstawowej nieprogramowalnej, a zatem niealgorytmicznej cechy. Wielu ludziom sama idea inteligentnej maszyny nie brzmi dobrze. Przedstawiono różne argumenty, aby uczynić nie do pomyślenia koncepcję inteligentnej maszyny myślącej. Niektórzy twierdzą, że myślenie musi koniecznie obejmować emocje i uczucia, a żaden komputer nie może nienawidzić, kochać ani wpadać w złość. Inni twierdzą, że inteligentne myślenie z konieczności pociąga za sobą oryginalność i żaden komputer nie może niczego stworzyć, jeśli nie zostanie zaprogramowany do tego z wyprzedzeniem, w którym to przypadku nie jest już oryginalny. Z tego punktu widzenia komputera nigdy nie można nazwać „inteligentnym”. Z drugiej strony wiele osób uważa, że ludzki mózg sam w sobie jest tylko maszyną, choć skomplikowaną. Tak więc, zgodnie z tezą Churcha/Turinga (zob. rozdział 9) komputer elektroniczny może w zasadzie symulować ludzki umysł i działać tak, jakby był inteligentny. Nazywa się

to słabym twierdzeniem AI. Idąc krok dalej, zwolennicy silnej sztucznej inteligencji twierdzą, że taki komputer byłby naprawdę świadomy. Więcej o tej debacie powiemy poniżej. Wydaje się jednak, że maszyna uważana za inteligentną musi przynajmniej być w stanie wykazywać podobne do ludzkiego zachowania intelektualne. W tym celu nie potrzebujemy go, aby chodzić, widzieć lub mówić jak człowiek, tylko rozumować i odpowiadać jak człowiek. Co więcej, jakiegokolwiek okazały się uzgodnione kryteria wywiadu, ktoś powinien być w stanie sprawdzić, czy maszyna-kandydat je spełnia. A kto, jeśli nie prawdziwy, inteligentny człowiek, ma kwalifikacje do przeprowadzenia takiego testu? To prowadzi nas do idei, że maszynę należy nazwać inteligentną, jeśli może przekonać przeciętnego człowieka, że pod względem intelektu nie różni się od innego przeciętnego człowieka.

Test Turinga

Jak możemy ustawić rzeczy, aby taki test był możliwy? W 1950 roku Alan Turing zaproponował następującą metodę, obecnie powszechnie nazywaną testem Turinga. Test odbywa się w trzech salach. W pierwszym jest przesłuchujący człowiek, nazwijmy go Alicją, w drugim drugi człowiek, a w trzecim komputer kandydat. Śledcza Alice zna pozostałą dwójkę tylko pod imionami Bob i Carol, ale nie wie, kto jest człowiekiem, a kto komputerem. Trzy pokoje są wyposażone w terminale komputerowe, a terminal Alicji jest połączony z terminalami Boba i Carol.



Teraz Alicja ma, powiedzmy, godzinę na ustalenie prawidłowej tożsamości Boba i Carol. Alicja może zadawać dowolne pytania lub wypowiedzi każdemu z nich, a komputer musi dołożyć wszelkich starań, aby oszukać Alicję, sprawiając wrażenie człowieka. Mówi się, że komputer zda test, jeśli Alicja nie wie, który z Bobów lub Carol jest naprawdę komputerem po upływie wyznaczonego czasu. (Właściwie wymagamy, aby komputer przeszedł szereg testów w jednej sesji, z różnymi przesłuchującymi, aby zminimalizować możliwość, że Alicja po prostu zgadnie, która jest która).

Spróbujmy wyczuć ogromną trudność z tym związaną. Zastanów się, jak inteligentny program musiałby zareagować na następujące pytania Alicji:

1. Czy jesteś komputerem?
2. Jaka jest godzina?
3. Kiedy zamordowano prezydenta Kennedy'ego?
4. Ile wynosi $2\ 276\ 448 \times 7\ 896$?

5. Czy białe mogą wygrać jednym ruchem z następującej pozycji w szachach: . . . ?

6. Opisz swoich rodziców.

7. Jak uderza cię następujący wiersz: . . . ?

8. Co myślisz o Charlesie Dickensie?

9. Jakie jest Twoje zdanie na temat programów eksploracji kosmosu, biorąc pod uwagę fakt, że miliony ludzi na całym świecie cierpią z głodu?

Zauważ, że zaprogramowany komputer musi być w stanie swobodnie rozmawiać w języku naturalnym, takim jak angielski. Rezygnujemy z potrzeby słyszenia i mówienia, a co za tym idzie łączy elektronicznych. Jednak rozumienie i syntetyzowanie języka uważane są za dwa z kamieni węgielnych ludzkiej inteligencji, a zatem są tutaj wymaganym warunkiem wstępnym. Pierwsze dwa pytania są dość proste. Odpowiedź na pierwsze powinno brzmieć „nie”, a na drugie program może korzystać z wbudowanego zegara, który jest obecny w większości komputerów. Pytanie (3) wymaga, aby program miał dostęp do bardzo dużej puli wiedzy, porównywalnej z wiedzą człowieka. Pytanie (4) wydaje się proste - komputery wydają się radzą sobie najlepiej z takimi pytaniami - ale mimo to jest subtelne, ponieważ komputer musi chwilę poczekać, zanim odpowie; natychmiastowa odpowiedź zdradziłaby jego tożsamość. Pytanie (5) wymaga oczywiście pewnej wiedzy o szachach, w tym umiejętności rozumowania o nowej pozycji. Chociaż odpowiedzi na pytania podobne do pytania (6) mogą być przygotowane z wyprzedzeniem, Alicja może odpowiedzieć na odpowiedź, podążając za pewnym nieoczekiwanym szczegółem, na który komputer może nie mieć gotowej odpowiedzi. Pytania (7), (8) i (9) są znacznie trudniejsze, ponieważ ilustrują potrzebę okazywania preferencji, opinii, stanowisk i emocji, wyjaśniania ich i odnoszenia ich do innych. Krótko mówiąc, konkurujący komputer musi być obdarzony prawdziwą osobowością. Jedną z trudności w zaprogramowaniu komputera do zdania testu Turinga jest duża różnorodność obszarów, do których może prowadzić rozmowa. Odnotowano pewne widoczne sukcesy w naśladowaniu ludzkiej konwersacji w bardzo ograniczonych obszarach dyskursu. Program o nazwie ELIZA, napisany w 1966 roku, był w stanie dać imponujący występ jako psychoanalityk. Tak imponujące, że ludzie byli znani z tego, że faktycznie „zwierzali się” w programie w sposób, w jaki nigdy nie zwierzyliby się człowiekowi! Oto kilka typowych wymian:

Osoba: Moja żona mówi, że często mam depresję.

ELIZA: Przykro mi słyszeć, że masz depresję.

Osoba: Prawda jest taka, że jestem bardzo nieszczęśliwa.

ELIZA: Myślisz, że przyjazd tutaj pomoże ci nie być nieszczęśliwym?

Osoba: Może rozmawiając z tobą dowiem się, dlaczego nienawidzę swojego brata.

ELIZA: Opowiedz mi więcej o swojej rodzinie.

Interesującą rzeczą dotyczącą ELIZY jest fakt, że nie można powiedzieć, że jest inteligentna w żaden rozsądny sposób. W rzeczywistości, pomimo pozornego wyrafinowania, ELIZA jest naprawdę głupia, więc o ile sprawia wrażenie rozważnej terapeutki, jest naprawdę sprytnym oszustem. Czasami po prostu skupia się na konkretnym słowie lub frazie, na które nauczono go zwracać uwagę, i odpowiada, wybierając, praktycznie losowo, jedną z niewielkiej liczby stałych odpowiedzi. Doskonałym tego przykładem jest odpowiedź „Opowiedz mi więcej o swojej rodzinie” wywołana słowem „brat”. Innym razem ELIZA po prostu zamienia nadchodzące zdanie w pytanie lub nieco pozbawione treści oświadczenie, jak w wymianie dotyczącej nieszczęścia danej osoby. Aby to zrobić, wykorzystuje prosty

mechanizm do odgadnięcia gruboziarnistej struktury zdań wejściowych. Dziwne rzeczy dzieją się, gdy naprawdę próbujesz przetestować inteligencję ELIZY, zamiast po prostu wylewać swoje problemy. W rzeczywistości, gdybyśmy powiedzieli: „Byłam siostrą w klasztorze w Birmie”, a nawet „Podziwiam siostrę Teresę”, program może równie dobrze odpowiedzieć taką samą nieistotną odpowiedzią: „Opowiedz mi więcej o swojej rodzinie”. Równie zabawna będzie jego odpowiedź na pytanie o programy eksploracji kosmosu. ELIZA oczywiście nie ma szans na zdanie testu Turinga. W rzeczywistości nie było to zamierzone. Motywacją stojącą za ELIZA było pokazanie, że łatwo jest wyglądać na inteligentnego, przynajmniej na krótką chwilę, sympatycznemu obserwatorowi i w wąskiej dziedzinie dyskursu. Bycie naprawdę inteligentnym to jednak zupełnie inna sprawa. Aby jeszcze bardziej docenić różnicę między prawdziwą inteligencją wymaganą do zdania testu Turinga, a płytką, ale zwodniczo zwodniczą naturą zdolności konwersacyjnych ELIZY, oto hipotetyczna wymiana zdań między przesłuchującą test Turinga Alice i naprawdę inteligentnym kandydatem, powiedzmy Bobem:

Alice: Co to jest zupczok?

Bob: Nie mam pojęcia.

Alice: Zupczok to latający wieloryb piszący powieści. Był starannie hodowany w laboratorium przez kilka pokoleń, aby zapewnić, że jego płetwy ewoluują w podobne do skrzydeł rzeczy, które umożliwiają mu latanie. Stopniowo uczono go także czytać i pisać. Posiada gruntowną wiedzę na temat literatury współczesnej i posiada umiejętność pisania powieści kryminalnych do publikacji.

Bob: Jakie to dziwne!

Alice: Czy myślisz, że zupczoki istnieją?

Bob: Nie ma mowy. Nie mogą.

Alicja: Dlaczego?

Bob: Z wielu powodów. Po pierwsze, nasze możliwości inżynierii genetycznej są dalekie od adekwatnych, jeśli chodzi o zamienianie płetw w skrzydła, nie wspominając o naszej niezdolności do spowodowania, aby 10-tonowe istoty z silnikami przeciwstawiły się grawitacji tylko poprzez machanie tymi rzeczami. Po drugie, część dotycząca pisania powieści nawet nie zasługuje na odpowiedź, ponieważ napisanie dobrej historii wymaga znacznie więcej niż umiejętności technicznych czytania i pisania. Cały pomysł wydaje się dość śmieszny. Nie masz nic ciekawszego do omówienia?

Aby przeprowadzić tę rozmowę, Bob, czy to człowiek, czy komputer, musi wykazywać bardzo wyrafinowane zdolności. Musi posiadać dużą wiedzę na konkretne tematy, takie jak wieloryby, latanie, pisanie powieści i grawitacja. Musi być w stanie nauczyć się zupełnie nowego pojęcia, przyjmując definicje i odnosząc je do tego, co już zna. Wreszcie, musi być w stanie wywnioskować z tego wszystkiego, na przykład fakt, że inżynieria genetyczna ma większe znaczenie dla zupczoków niż, powiedzmy, matematyka czy chińska filozofia. (W tym konkretnym przypadku musi też mieć poczucie humoru.)

Silna sztuczna inteligencja i chiński pokój

Załóżmy, że stworzono program komputerowy, który może przejść test Turinga. Zgodnie z silnym twierdzeniem AI, ten program byłby uważany za inteligentny i świadomy. Słynny eksperyment myślowy, zwany chińskim argumentem pokojowym, próbuje obalić tego rodzaju twierdzenia. Oto jak to działa. Rozważ zmodyfikowaną wersję programu, która rozmawia po chińsku, a nie po angielsku. Zasadniczo jest to zestaw precyzyjnych instrukcji dla komputera i można go przetłumaczyć na serię podobnych instrukcji w języku angielskim, aby człowiek mógł je wykonać. (Byłoby to długie i żmudne,

ale w zasadzie nadal możliwe, a chińskie części wejścia/wyjścia mogą zawierać „głupie” rzeczy, takie jak „jeśli patrzysz na ten chiński znak i kwadrat numer 159 w Twoim notatniku zawiera 1, napisz 0 w kwadracie 243 i przejdź do następnego znaku w prawo.”) Przypuśćmy teraz, że książka zawierająca pełny zestaw instrukcji angielskich jest przekazywana osobie, która rozumie i mówi po angielsku, ale nie po chińsku, a następnie zostaje zamknięta w pomieszczenie ze szczeliną do komunikacji ze światem zewnętrznym. Pytania napisane po chińsku są wkładane przez szczelinę, a osoba zamknięta w pokoju postępuje zgodnie z instrukcjami zawartymi w księdze, aby uzyskać odpowiedź, ponownie po chińsku. Ponieważ dana osoba podąża za oryginalnym programem komputerowym, jej odpowiedzi należy uznać za inteligentne zgodnie z testem Turinga. Najwyraźniej jednak osoba przebywająca w tak zwanym chińskim pokoju nie rozumie treści komunikatów, a jedynie postępuje zgodnie z instrukcjami zawartymi w jego książce. Tak więc chińskiego pokoju i jego użytkownika nie można opisać jako świadomych, czy nawet inteligentnych, a zatem oryginalny program też nie! Napisano wiele artykułów i książek, aby wesprzeć jedną lub drugą stronę tej debaty, ale nie będziemy ich tutaj dalej omawiać. Podczas gdy filozofowie debatują i spierają się, badacze sztucznej inteligencji wciąż próbują opracować naprawdę inteligentne programy. Należy jednak zauważyć, że program, który jest w stanie przejść test Turinga, nie rozwiąże problemu; raczej posłuży raczej do jej zintensyfikowania. Gdyby taki program upierał się, że jest świadomy, czy uwierzyłbyś w to? Jeśli o to chodzi, możesz zadać to samo pytanie także ludziom: czy jesteśmy naprawdę inteligentni, czy po prostu symulujemy inteligencję dzięki programowaniu naszych mózgów?

Grać w gry

Test Turinga wydaje się zapewniać wystarczający warunek, aby komputer posiadał pełną ludzką inteligencję. Jednak większość badaczy sztucznej inteligencji nie postawiła sobie za cel pisania programów, które mogą zdać test Turinga, ponieważ obejmuje wiele rzeczy, które nie są bezpośrednio związane z czystą inteligencją. Na przykład, aby zdać test Turinga, komputer musiałby zostać zaprogramowany tak, aby ukrywał swoje nadludzkie umiejętności, jak widać w pytaniu (4) powyżej. Jest to podobne do tego, co wydarzyło się w innych dziedzinach. Na przykład przez wiele lat ludzie próbowali osiągnąć „sztuczny lot”, próbując naśladować ptaki. Sukces osiągnięto dzięki zwróceniu się ku zupełnie innym metodom, a nowoczesnym samolotom przypisujemy zdolność latania, nawet jeśli nikogo nie oszukają, by sądził, że są ptakami. Współczesne badania nad sztuczną inteligencją można z grubsza podzielić na analizę i syntezę. Celem pierwszego jest zrozumienie natury inteligencji; często wiąże się to z badaniem tego, w jaki sposób ludzie uczą się, rozumują, dedukują i tworzą plany oraz powielają te umiejętności w programach komputerowych. Ten kierunek badań ma zatem wiele wspólnego z dziedzinami takimi jak psychologia i neurobiologia, oprócz matematyki i algorytmiki. Drugie podejście ma na celu zbudowanie „inteligentnych” programów komputerowych, które mogą wykonywać przydatne zadania. To, jak podobne są do ludzkiej inteligencji, jest mniej ważne. Początkowe prace w latach 50. i 60. miały na celu zbudowanie ogólnych metod rozwiązywania różnych problemów szczegółowych. Stwierdzono, że jest to znacznie trudniejsze niż początkowo sądzono, a później prace skierowano do węższych dziedzin, aby wykorzystać specjalistyczną wiedzę związaną z określonymi tematami. Granie w gry to jeden z wyspecjalizowanych obszarów, w których badania nad sztuczną inteligencją osiągnęły znaczące wyniki. Wydaje się właściwe rozpoczęcie dyskusji od krótkiego biuletynu informacyjnego.

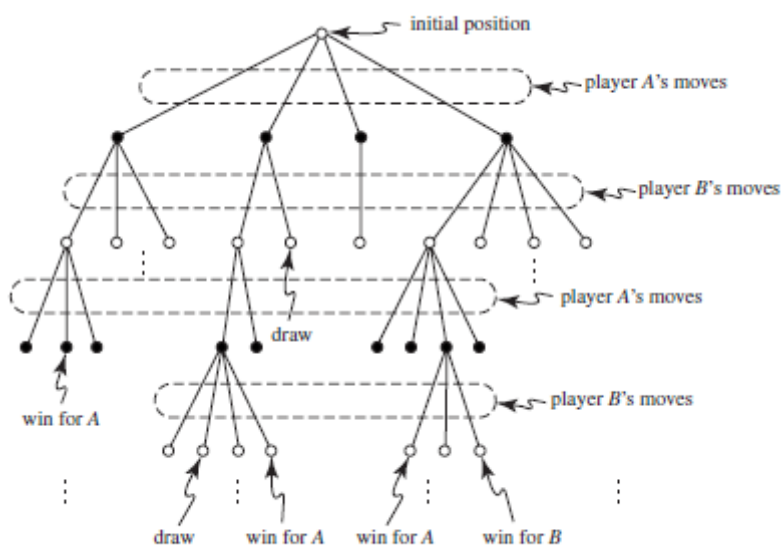
* W 1979 roku program komputerowy pokonał mistrza świata w tryktraku. (To nie uczyniło programu nowym mistrzem, ponieważ gra nie była rozgrywana w oficjalnym turnieju, ale mimo wszystko wygrana była wygraną).

* W 1994 roku mistrz świata w warcaby Marion Tinsley zrezygnował z tytułu na rzecz programu komputerowego Chinook. Ten program wygrał mistrzostwa Man-Machine Checkers Championship,

które odbywają się między najlepszym graczem człowiekiem a najlepszym graczem programu. (Ten specjalny tytuł został stworzony, aby można było mieć zarówno mistrza świata ludzi, jak i komputerowego mistrza świata.)

* W 1997 roku mistrz świata w szachach, Gary Kasparow, przegrał z komputerem Deep Blue od 3,5 do 2,5 w sześciu grach. Ten mecz nastąpił po wygranym 4-2 przez Kasparowa w 1996 roku. Na początku 2003 roku Kasparow zebrał rundę i zremisował 3-3 z Deep Junior, programem, który zdobył tytuł mistrza świata w szachach komputerowych w 2002 roku.

Chociaż programy mogą teraz grać w doskonałe gry w tryktrak, warcaby i szachy, nadal nie są doskonałe. Istnieją inne gry, takie jak Go, w których komputery wcale nie radzą sobie tak dobrze. Dlaczego programy nie zawsze mogą grać w idealną grę? Dlaczego komputer nie może wykonać wszystkich możliwych ruchów i zawsze wykonać najlepszy? Odpowiedź tkwi w drzewach łownych, o których wspomniano w Części 2.



Na przykład w kółko i krzyżyk (kółko i krzyżyk) nie ma trudności. Pierwszy gracz ma dziewięć możliwych ruchów, na które jego przeciwnik może odpowiedzieć na jeden z ośmiu sposobów, na który pierwszy może odpowiedzieć na jeden z siedmiu itd. Drzewo gry składa się zatem z korzenia z dziewięcioma potomstwem, z których każdy ma osiem potomstwo i tak dalej. Niektóre węzły w tym drzewie są terminalami, co oznacza, że reprezentują albo wygraną jednego z graczy, albo pełną planszę bez wygranej. W każdym razie dowolna sekwencja dziewięciu ruchów prowadzi do węzła końcowego. Drzewo ma zatem maksymalną głębokość 9, z maksymalnym stopniem zewnętrznym 9 u nasady. W sumie jest nie więcej niż $9!$, czyli 362,880 możliwości sprawdzenia; stąd program, który można łatwo napisać, aby wydajnie grać perfekcyjnie w kółko i krzyżyk. Z drugiej strony w szachach historia jest zupełnie inna. Białe mają 20 możliwych pierwszych ruchów, a średnia liczba możliwych następnych ruchów z dowolnej pozycji w szachach wynosi około 35. Tak więc odchylenie drzewa wynosi średnio około 35. Głębokość drzewka to liczba ruchów (dwukrotność liczby rund gry), która z łatwością może osiągnąć 80 lub 100. Oznacza to, że liczba możliwości sprawdzenia w typowej grze może wynosić 35100. W rozdziale 7 przedstawiliśmy kilka takich liczb i pamiętamy, że 35100 to wiele, wiele rzędów wielkości większa niż liczba protonów we wszechświecie, czyli liczba nanosekund od Wielkiego Wybuchu. W konsekwencji, nawet jeśli zignorujemy księgowość i pamięć zaangażowaną w brutalną podróż przez wszystkie możliwe ruchy i założymy, że każdy z nich można przetestować w ciągu

jednej nanosekundy, żaden program nigdy nie zagra w perfekcyjne szachy. Liczby dla warcabów nie są aż tak duże, ale idealne warcaby również nie wchodzą w rachubę. Liczby dla Go są nawet wyższe niż dla szachów; na przykład liczba możliwych ruchów w każdej pozycji wynosi zwykle około 200, a nie 35, a gra trwa około 300 ruchów! Jak zatem działają dobre programy szachowe? Cóż, rzeczywiście przeprowadzają masowe przeszukiwanie dużych części drzewa gry, ale używają także heurystyk lub reguł kciuka. W kontekście gier heurystyka jest używana do pomocy w podjęciu decyzji, które części drzewa gry będą brane pod uwagę przy próbie wybrania dobrego ruchu. Nietypowe wyszukiwanie heurystyczne wykorzystuje intuicyjne reguły wprowadzone do programu przez programistę, aby zignorować pewne części drzewa gry. Na przykład można zdecydować, że jeśli w ciągu ostatnich pięciu ruchów nic się nie zmieniło w czterokwadratowym sąsiedztwie pewnego pionka, to ten pionek nie zostanie przesunięty, a zatem wyszukiwanie może zignorować wszystkie części drzewa znajdujące się poniżej odpowiedni węzeł. Taka zasada może okazać się bardzo wnikliwa — z pewnością skutkuje rozważeniem mniejszych drzew — ale oczywiście może to kosztować grę; Gary Kasparow mógł awansować tego samego pionka, aby wygrać w trzech ruchach. Jest to bardzo prosty przykład, a heurystyki zawarte w prawdziwych programach do gry w szachy są zwykle znacznie bardziej wyrafinowane. Niemniej jednak są to heurystyki, a ich użycie zwiększa prawdopodobieństwo, że możemy przegapić najlepszy ruch.

Więcej o heurystyce

Dobrym sposobem na wyjaśnienie natury wyszukiwania heurystycznego jest rozważenie osoby, która zgubiła soczewkę kontaktową. Jedną z możliwości jest przeprowadzenie wyszukiwania na ślepo, pochylając się i obmacując obiektyw w sposób przypadkowy. Inną możliwością jest przeszukiwanie systematyczne, polegające na ciągłym poszerzaniu przeszukiwanego obszaru, w sposób metodyczny i zorganizowany (powiedzmy, w coraz większych kręgach wokół punktu centralnego). To poszukiwanie w końcu się powiedzie, ale może być bardzo czasochłonne. Trzecią możliwością jest poszukiwanie analityczne, w którym obliczane są dokładne równania matematyczne rządzące opadaniem soczewki, biorąc pod uwagę wiatr, grawitację, tarcie powietrza i dokładną topografię, napięcie i fakturę powierzchni. To również, jeśli zostanie zrobione poprawnie, gwarantuje sukces, ale z oczywistych powodów jest to niepraktyczne. W przeciwieństwie do tych metod, większość z nas podchodziłaby do problemu za pomocą wyszukiwania heurystycznego. Najpierw określilibyśmy przybliżony kierunek upadku i zgadywali, jaką odległość soczewka mogła pokonać, spadając, a następnie ograniczylibyśmy poszukiwania do wynikowego obszaru. Tutaj oczywiście wyszukiwanie może się nie powieść, ale wydaje się, że jest całkiem spora szansa, że się uda. (Istnieje oczywiście piąte podejście, leniwe wyszukiwanie, które wymaga znalezienia najbliższego optyka i wykonania nowego obiektywu...) Główną wadą heurystyk jest to, że nie gwarantują sukcesu. Zawsze jest możliwe, że w konkretnym przypadku zawiedzie praktyczna reguła. Jeśli chodzi o korzyści, oprócz możliwości drastycznego skrócenia czasu pracy, heurystyki są zwykle podatne na ulepszenia lub wymianę, gdy lepiej poznajemy dany problem i sposoby, w jakie ludzie go radzą. Jednak w kontekście algorytmiki najważniejszą cechą heurystyki jest to, że jej wykonanie nie podlega analizie. Możemy zdecydować się na heurystykę szachową, włączyć ją do naszego programu gry w szachy i od tej pory będziemy mogli oceniać jej wykonanie tylko przez obserwację. Może rozegrać 100 doskonałych gier, zanim odkryjemy, że ma poważną słabość, która kiedyś znana przeciwnikowi dramatycznie pogarsza jego grę. W pewnym sensie stosowanie heurystyki przypomina trochę rzucanie monetami, ponieważ niekoniecznie uwzględniamy wszystkie możliwości i w konsekwencji możemy przegapić dobre rozwiązanie. W rozdziale 11 zobaczyliśmy, jak piły można ulepszyć, podążając za kaprysmi losowych rzutów monetą. Innymi słowy, przestrzeń poszukiwań wszystkich możliwości została zmniejszona, a niektóre kierunki pozostały niezbadane. Byliśmy więc skłonni nazwać liczbę „pierwszą”, chociaż nie sprawdziliśmy wszystkich możliwych świadków jej niepierwotności. Również tam sukces nie jest gwarantowany; stąd kuszące

jest, aby postrzegać rzucanie monetą jako ślepą heurystykę, rodzaj pozbawionej intuicji zasady kciuka. Jest jednak zasadnicza różnica. W dziedzinie algorytmów probabilistycznych analiza zastępuje intuicję. Stosując starannie zdefiniowane zestawy możliwych do zignorowania możliwości i stosując randomizację, aby zdecydować, które faktycznie zignorować, jesteśmy w stanie przeanalizować prawdopodobieństwo wystąpienia rygorystycznego sukcesu, składanie precyzyjnych oświadczeń na temat wydajności algorytmu. Z heurystykami zazwyczaj nie możemy.

Chociaż probabilistyczny algorytm testowania pierwszości (a właściwie każdy algorytm, nawet prosta procedura sortowania) zdecydowanie wygląda na inteligentny — w rzeczywistości robi to znacznie lepiej niż przeciętny człowiek — nie uważamy go za naprawdę inteligentny. Chociaż jego konstrukcja mogła wymagać intuicyjnej pomysłowości ze strony projektanta, jego działanie nie opiera się na intuicji i można je wytłumaczyć analitycznie. Z kciuka, których wyników nie możemy przewidzieć ani przeanalizować. Jest to zatem jedna z możliwości interpretacji naszego poczucia, że prawdziwa inteligencja jest nieprogramowalna; po prostu zamień nonprogrammable na nonanalyzable. AI charakteryzuje się programami opartymi na regułach, które wydają się pomocne, ale których przydatność nie została dokładnie przeanalizowana. Ta prymitywna próba zdefiniowania sztucznej inteligencji przez jakąś właściwość powstałych programów nie jest całkiem sprawiedliwa. Większość ludzi wolałaby zdefiniować to tematycznie. Co więcej, niektóre kierunki badań nad sztuczną inteligencją stały się ostatnio dość precyzyjne i analityczne. W wielu przypadkach stosowane heurystyki są czymś więcej niż tylko wykształconymi domysłami; w rzeczywistości opierają się na dobrze zdefiniowanych modelach matematycznych i wzorach. Na przykład w skomputeryzowanym widzeniu heurystyka stosowana do wykrywania ruchu i rozumienia stereoskopowych par obrazów jest oparta na skomplikowanej matematyce i formułuje się mocno poparte przypuszczenia dotyczące losowego zachowania się programów wynikowych wejścia.

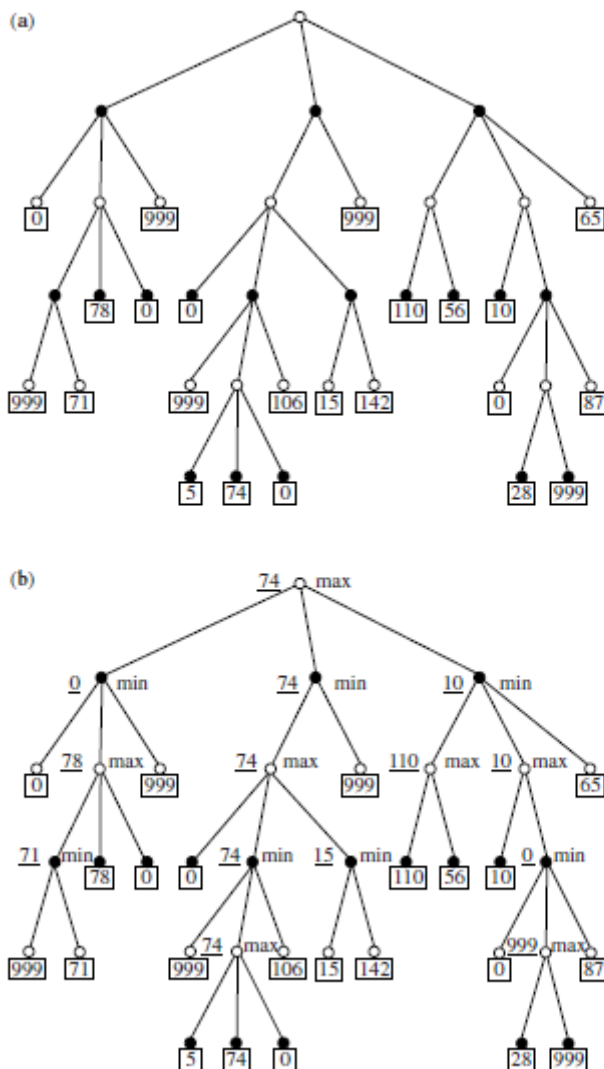
To samo można powiedzieć o planowaniu ruchu w robotyce. W takich przypadkach algorytmy heurystyczne są bliższe algorytmom aproksymacyjnym z Części 7 niż algorytmom probabilistycznym, ponieważ stosowane heurystyki są zwykle domniemane albo w celu zagwarantowania średnio dobrego rozwiązania, albo niezwykle rzadko dają złe rozwiązanie. Możemy zatem nazwać otrzymane algorytmy algorytmami aproksymacji hipotetycznej — brakuje tylko dowodu własności hipotetycznej. Oczywiście po znalezieniu dowodu nie spodziewalibyśmy się, że dyscyplina AI zrezygnuje z algorytmu tylko dlatego, że jego zachowanie zostało dokładnie przeanalizowane

Ocena i wyszukiwanie

Opis heurystyk w skomputeryzowanych szachach również był zbyt uproszczony. W rzeczywistości dzieje się o wiele więcej niż kilka prostych zasad, które powodują, że program ignoruje części drzewa gry. Na przykład musi istnieć sposób oceny jakości pozycji w drzewie. Jako prosty przykład rozważmy węzeł, dla którego zdecydowano zignorować wszystkie z wyjątkiem dwóch możliwych następnych ruchów białych. Załóżmy, że pierwszy z nich może ostatecznie doprowadzić do 10 węzłów końcowych, z których trzy reprezentują wygraną białych, jeden wygraną czarnych, a sześć remisów, a w pozostałych liczbach nie będzie 10, 3, 1 i 6, ale powiedzmy 8, 4, 3 i 1. Jak powinien być program (który gra Białymi) porównać te sytuacje, aby wybrać następny ruch? Problem staje się znacznie bardziej dotkliwy, gdy nawet w rozważanych kierunkach chcemy przerwać poszukiwania na pewnej głębokości. W takim przypadku osiągnięto najbardziej odległe pozycje w wyszukiwaniu nie dają konkretnych informacji o wygranej/przegranej/remisie; mamy tylko informacje, które są ukryte w samej konfiguracji płytki. W takim przypadku funkcja oceny jest znacznie mniej oczywista. Problem oceny sytuacji i przypisywania im wartości liczbowych pomagających w podjęciu decyzji jest jednym z głównych wyzwań programowania heurystycznego i nie ogranicza się tylko do grania w gry. Rozważ program do przeprowadzania zabiegów medycznych i diagnoz. Tutaj również istnieje drzewo, którego węzły

reprezentują kombinacje symptomatycznych problemów i pytań skierowanych do pacjenta, przy czym węzły końcowe reprezentują ostateczne diagnozy. Tutaj też drzewo jest ogromne; ponadto niektóre węzły odpowiadają różnym rodzajom testów medycznych, które niosą ze sobą własne ryzyko. W związku z tym musi nastąpić poszukiwanie heurystyczne, z obserwowalnymi objawami pacjenta i jego odpowiedziami na pytania określające kierunki, które będą realizowane i testy, które zostaną wykonane. Równie trudny jest tu problem ewaluacji, który określa, jak istotny jest dany zestaw możliwości dla pożądanej ostatecznej diagnozy.

Niezależnie od tego, czy w szachach, w medycynie, czy gdzie indziej, po zdefiniowaniu funkcji oceny dla drzewa poszukiwań, nadal pozostaje problem wykorzystania wartości węzłów w drzewie do efektywnego przeszukiwania jego odpowiednich części. Tutaj praca nad sztuczną inteligencją zaowocowała wieloma potężnymi metodami wyszukiwania, które mają wiele zastosowań także w algorytmice nieheurystycznej. Wiele z nich opiera się na podstawowej strategii wyszukiwania zwanej metodą minimax, którą najlepiej wyjaśnić w ramach drzew gry.



Rysunek (a) przedstawia część drzewa gry dla gry dwuosobowej, np. w kółko i krzyżyk lub szachy, wraz z wcześniej ustalonymi wartościami dla każdego węzła końcowego. (Końcowe węzły na rysunku są albo rzeczywistymi pozycjami końcowymi gry, albo pozycjami pośrednimi, poza którymi zdecydowaliśmy się nie szukać.) Korzeń reprezentuje pozycję na planszy, na której gracz Alicja ma iść, a wartości

reprezentują siłę pozycji z punktu widzenia Alicji. Tak więc 999 oznacza wygraną Alicji, a 0 wygraną Boba. Wartości pośrednie odzwierciedlają względne szanse na wygraną Alicji z odpowiednich pozycji, zgodnie z pewną heurystyczną funkcją oceny. Dla Alicji należy wybrać jeden możliwy ruch, który zmaksymalizuje jej zdolność do wygrania. Podstawową ideą jest wielokrotne propagowanie wartości w górę drzewa, zaczynając od dołu. Jeśli bieżący węzeł rodzicielski reprezentuje ruch Alicji, wtedy maksymalna z wartości węzłów potomnych jest dołączona do rodzica, a jeśli reprezentuje ruch Boba, przyjmowana jest wartość minimalna. Uzasadnienie tego jest jasne: zakłada się, że Bob gra w rozsądny sposób, a zatem dołoży wszelkich starań, aby zmaksymalizować własne szanse na wygraną, co oznacza, że będzie starał się zminimalizować szanse Alicji na to. Innymi słowy, Alicja powinna wykonać najlepszy ruch (tzn. ten z maksymalną wartością) przy założeniu, że po nim Bob wykona swój najlepszy ruch (tzn. ten z minimalną wartością funkcji Alicji), po czym Alicja zrobi jak najlepiej w następnym ruchu i tak dalej. Rysunek (b) przedstawia drzewo z wprowadzonymi wartościami min i max. Proces ten można skrócić obserwując, że w niektórych przypadkach nawet częściowa informacja o wartościach potomstwa drugiego pokolenia węzła (czyli jego wnuków) jest wystarczająca do określenia jego wartości końcowej. Na przykład na rysunku poniżej nie ma sensu oceniać skrajnego prawego z pięciu poddrzew drugiej generacji, ponieważ minimalna z trzech skrajnych lewych poddrzew wynosi 42, czyli więcej niż 27, czyli znana wartość po prawej stronie. Minimum 27 i jakakolwiek możliwa wartość korzenia najbardziej prawego drzewa nie będzie większa niż 27, a zatem maksymalizacja, która zostanie przeprowadzona w celu określenia wartości u korzenia, nieuchronnie doprowadzi do wartości 42. Podobny przypadek ma miejsce, gdy dany węzeł reprezentuje ruch Boba, a nie ruch Alicji, ale z maksymalną i minimalną zmianą miejsc. Znalaziono metody przemierzania drzewa podczas procesu propagacji w taki sposób, aby wykorzystać te oszczędności. Jednym z bardziej znanych z nich, który nie zostanie tutaj opisany, jest przycinanie alfa-beta. Podsumowując, wyszukiwanie heurystyczne składa się z:

1. heurystyki zawarte w funkcji wartościowania;
2. zasady określające, jak głęboko będzie wyglądać perspektywa z dowolnej pozycji (zazwyczaj jest to funkcja danej pozycji); oraz
3. sprawna procedura propagacyjna do określania wartości i faktycznego dokonywania wyborów.

Reprezentacja wiedzy

Aby osiągnąć inteligencję algorytmiczną, potrzebujemy czegoś więcej niż tylko heurystyk. Musimy znaleźć sposoby na przedstawienie wiedzy, którą manipulują inteligentne algorytmy. Jeśli części kontrolne programu AI są szczególne, oparte na „miękkim” pojęciu heurystyki, to części danych również są szczególne, oparte na „miękkim” pojęciu wiedzy. To, że dwa razy cztery to osiem i że Francja jest w Europie to wiedza, ale to samo dotyczy faktu, że wszystkie żyrafy mają długie szyje, że Alan Turing był genialny i że naukowcy, którzy nie publikują, giną. Ale co to jest „długi”, a co „genialny” i czy „zginąć” znaczy dosłownie? Co więcej, w jaki sposób przedstawiamy takie fakty w naszych umysłach lub w naszych algorytmicznych bazach wiedzy i jak je wykorzystujemy? Żaden program nie może być oznaczony jako inteligentny, czy to ten, który działa w wąskiej dziedzinie, takiej jak szachy, bloki i piramidy, czy też jako uniwersalny kandydat do zdania testu Turinga, chyba że ma odpowiedni mechanizm przechowywania, wyszukiwania i manipulowania wiedzą, umiejętności. Problem reprezentacji wiedzy jest rzeczywiście jednym z centralnych zagadnień sztucznej inteligencji. Trudność polega na tym, że wiedza to nie tylko duży zbiór faktów, ale także wiele zawitych relacji między nimi. Relacje te mają wiele aspektów i atrybutów, które z kolei tworzą inne, wyższego poziomu relacje z innymi elementami wiedzy. Niewiele wiemy o tym, jak sami przechowujemy i manipulujemy ogromnymi ilościami wiedzy zgromadzonymi w ciągu naszego życia. W rzeczywistości istnieją dowody

na to, że ludzka baza wiedzy jest dynamiczna i asocjacyjna i niekoniecznie działa w sposób sugerujący działanie współczesnych komputerów cyfrowych. Wiele modeli wiedzy zostało zaproponowanych do wykorzystania przez inteligentne programy. Niektóre opierają się na czystych koncepcjach informatycznych, takich jak relacyjne lub hierarchiczne bazy danych, a inne na formalizmach logicznych, takich jak rachunek predykatów lub logika modalna. Niektóre języki programowania, takie jak LISP i PROLOG, które są omówione w Części 3, są bardziej odpowiednie do manipulowania wiedzą niż inne. PROLOG, na przykład, jest całkiem trafny, jeśli chodzi o elementy wiedzy dotyczące prostych relacji, takich jak „Jan jest ojcem Marii”, a nawet „rodzic dowolnego przodka Marii jest również przodkiem Maryi”. Jednak poza małą, dobrze zdefiniowaną domeną dyskursu wymagane relacje stają się znacznie bardziej skomplikowane, a takie modele stają się zdecydowanie nieodpowiednie. Odzyskiwanie tych elementów wiedzy, które są istotne dla jakiejś decyzji, którą program musi podjąć, staje się ogromnym zadaniem. Nie znaleziono zatem „właściwego” modelu algorytmicznej reprezentacji wiedzy

Systemy oparte na wiedzy

Jednym ze sposobów podejścia do problemu są zasady produkcji. Prosta reguła produkcyjna może stwierdzać, że jeśli X jest firmą, a Y nią zarządza, to Y pracuje dla X. Bardziej złożona reguła, prawdopodobnie odnosząca się do programu rozumienia sceny, stwierdzałaby, że za każdym razem, gdy zostaną znalezione trzy linie spotykające się w jednym punkcie wtedy jest możliwe, że reprezentują róg trójwymiarowego sześcianu, chyba że jeden z kątów jest mniejszy niż 45° , a inny większy niż 90° . Niektóre programy specjalnego przeznaczenia zostały nazwane systemami opartymi na wiedzy lub systemami eksperckimi², ponieważ opierają się na regułach stosowanych przez eksperta w celu rozwiązania konkretnego problemu. Typowy system oparty na wiedzy jest konstruowany poprzez wypytanie eksperta o sposoby, w jakie wykorzystuje on wiedzę ekspercką w rozwiązywaniu danego problemu. Pytający (człowiek), czasami nazywany inżynierem wiedzy, próbuje odkryć i sformułować reguły używane przez eksperta, a system oparty na wiedzy następnie wykorzystuje te reguły do kierowania poszukiwaniem rozwiązania danego przypadku problemu. Sercem systemu opartego na wiedzy jest zestaw reguł podany w pewnym formacie oraz odpowiadający mu mechanizm wyszukiwania do znajdowania reguł, które mają zastosowanie. Powstałe systemy nazywane są również systemami produkcji lub systemy oparte na regułach. Oparte na wiedzy systemy o akceptowalnym poziomie wydajności zostały zbudowane do przeprowadzania ograniczonych form diagnozy medycznej, przydzielania zasobów, takich jak bramki lotnisk do przylatujących lotów, planowania i harmonogramowania operacji dla statków kosmicznych oraz planowania logistyki. Takie systemy oszczędzają wiele milionów dolarów korporacjom, które z nich korzystają. W rzeczywistości Amerykańska Agencja Projektów Badań Obronnych (DARPA), przez wiele lat główna agencja finansująca badania nad sztuczną inteligencją w Stanach Zjednoczonych, stwierdziła, że system planowania i harmonogramowania transportu użyty podczas wojny w Zatoce Perskiej w 1991 r. zawiązką zwrócił całkowitą kwotę DARPA. inwestycja w sztuczną inteligencję przez okres 30 lat! Musimy jednak zdać sobie sprawę, że oprócz polegania na wyszukiwaniu heurystycznym, reguły kontrolujące wyszukiwanie są tworzone przez kwestionowanie ekspertów, którzy nie zawsze działają według sztywnych reguł. Szanse na nieoczekiwane, być może katastrofalne zachowanie w systemie opartym na wiedzy są zatem nie do pominięcia. Niektórzy ludzie ujmują to w ten sposób: „Czy chciałbyś, aby zaopiekował się tobą skomputeryzowany oddział intensywnej terapii, który został zaprogramowany zgodnie z paradygmatem systemu opartego na wiedzy?” W rzadkich przypadkach urządzenie może podać niewłaściwy lek lub w niewłaściwym momencie zamknąć kluczowy zawór. Jego zachowanie rządzi się regułami, które zostały sformułowane podczas wywiadów z lekarzami ekspertami, którzy niekoniecznie muszą działać w nietypowym przypadku według dobrze sformułowanych reguł. (Oczywiście podobne rzeczy miały miejsce w przypadku programów

nieheurystycznych z powodu nieodpowiednich praktyk inżynierii oprogramowania, jak widzieliśmy w przypadku Therac-25 w Rozdziale 5.) Możliwe jest podjęcie prób poddania krytycznych systemów opartych na wiedzy precyzyjnym technikom analitycznym, co umożliwi nam formalną weryfikację, czy określone właściwości bezpieczeństwa mają zastosowanie w przypadku zastosowania zasad leżących u ich podstaw. Ponieważ, jak wyjaśniono w rozdziale 10, właściwości związane z bezpieczeństwem zwykle zakładają, że złe rzeczy się nie wydarzą, wydaje się, że systemy oparte na regułach byłyby podatne na takie podejście.

Wiedza w zakresie uczenia się, planowania i dedukcji

Problem reprezentacji wiedzy jest rzeczywiście fundamentalny dla mechanizacji wszelkiego rodzaju inteligencji i pojawia się ponownie w uczeniu się, planowaniu i dedukcji. Oto kilka typowych przykładów, które dodatkowo ilustrują potrzebę wyrafinowanej reprezentacji wiedzy. O żadnym programie nie można powiedzieć, że jest naprawdę inteligentny, jeśli nigdy się nie nauczy, jest na zawsze skazany na powtarzanie poprzednich błędów i nigdy nie staje się lepszy. Rozważ program w warcabach, który się uczy. Można założyć, że znasz zasady, a następnie po prostu uczysz się, jak unikać błędów, które popełnia, gdy gra więcej gier. Jednak nawet tutaj algorytmiczny punkt widzenia przedstawia poważne problemy reprezentacyjne. Czy po prostu robimy listę pozycji i ruchów, które okazały się złe, i odtąd wielokrotnie je przeglądamy, aby uniknąć ponownego popełnienia tych samych błędów? A może staramy się zapamiętać i zaktualizować bardziej ogólne zasady dobrej gry, które posłużą do modyfikacji heurystyki programu? Te pytania stają się tym trudniejsze, gdy temat jest szerszy. Jak uczą się dzieci? Jak przedstawiają wiedzę, która pozwala im rozpoznawać znajome twarze lub syntetyzować wiadomości, których nigdy wcześniej tak naprawdę nie słyszeli? W jaki sposób dorosły zapamiętuje i odzyskuje wiedzę, która pozwala mu nauczyć się pisać wypracowanie, organizować finanse osobiste lub przystosować się do nowego środowiska? Wcześniej wspomnieliśmy o dedukcji jako o jednym z rodzajów inteligencji wymaganych do zdania testu Turinga. Zostały napisane programy, które dość dobrze radzą sobie z dowodzeniem twierdzeń w geometrii szkoły średniej. Potrafią gromadzić istotne fakty i wyciągać z nich nowe. Jak to jest zrobione? Czy te programy po prostu przechowują listę znanych twierdzeń i kilka logicznych reguł wnioskowania (takich jak „jeśli P jest prawdziwe, a P zawsze implikuje Q, to Q też jest prawdziwe”)? A może są zaznajomieni z bardziej złożonymi łańcuchami rozumowania, które są nastawione na tworzenie interesujących twierdzeń? Ponownie, pytania te stają się tym trudniejsze, gdy wiedza związana z pożądanymi dedukcjami nie ogranicza się do wąskiej dziedziny. W jaki sposób wiedza pozwala na wyciągnięcie wniosku, że prawdopodobnie jutro będzie padać, wybór odpowiedniej szkoły dla dziecka lub udowodnienie naprawdę głębokiego twierdzenia w topologii algebraicznej?

Większość naszych dotychczasowych dyskusji dotyczyła wewnętrznych działań inteligentnych programów. Potrzeba mówienia, chodzenia lub widzenia była wielokrotnie rezygnowana, koncentrując się zamiast tego na zrozumieniu, uczeniu się i dedukcji. Jednak bardziej ogólny pogląd na inteligentne maszyny wymaga, aby posiadały one również zdolność do fizycznego naśladowania ludzi. W swojej najbardziej ogólnej formie muszą być w stanie rozumieć i syntetyzować ludzką mowę, a także wykonywać zadania fizyczne w duchu klasycznego robota. Również tutaj problem reprezentacji wiedzy jest problemem fundamentalnym. Niektóre systemy AI mogą rozumieć proste widoki bloków i piramid przy użyciu odpowiedniego wizualnego sprzętu sensorycznego, a inne mogą rozumieć wyraźnie mówiony elementarny angielski przy użyciu sprzętu audio. Jak oni to robią? Czy programy wizyjne po prostu znają każdą możliwą kombinację lokalizacji odpowiednich obiektów, czy też rozpoznają różne konfiguracje skrzyżowań linii i wykorzystują reguły do wywnioskowania z nich ogólnego układu? Co się stanie, jeśli zostanie wprowadzony nowy rodzaj obiektu, powiedzmy cylinder? Czy programy do rozpoznawania mowy mają bazę danych zawierającą profile fal dla wszystkich możliwych wymowy

każdego słowa? A może mają wbudowane reguły, które pozwalają im łączyć wypowiedziane segmenty słów w całość? Ponownie, jeśli domena jest znacznie szersza, jak te, które napotyka człowiek oglądający nowe otoczenie lub słuchający bogatej i różnorodnej rozmowy, sprawy stają się znacznie bardziej skomplikowane. Jak ludzie rozumieją różnorodność kolorów, linii i kształtów, które składają się na wnętrze salonu? Jak rozpoznają ruch i odległość od obserwowania rzeczy dwójkiem oczu w krótkim czasie i jak rozumieją obce akcenty? Umiejętność planowania to kolejna inteligentna umiejętność. Niektóre roboty, które działają w bardzo ograniczonym otoczeniu, są w stanie zaplanować sekwencję ruchów, która zaprowadzi je do celu. Jak oni to robią? Czy po prostu przeszukują możliwości, heurystycznie lub w inny sposób? A może wykorzystują bardziej subtelną wiedzę, która umożliwia im planowanie w celu osiągnięcia bardziej ogólnych celów? Ponownie, szersze domeny znacznie utrudniają sprawę. W jaki sposób osoba planuje podróż, nakreśla plan zakończenia roku z dodatnim saldem lub obmyśla strategię wygrania wojny? Jeśli inteligencja jest źródłem życia, jak mówi Księga Przysłów, to problem reprezentacji wiedzy jest z pewnością jej kamieniem węgielnym, a stwierdzenie, czy możemy tchnąć życie — że tak powiem — w komputer wymaga znalezienia odpowiedniego rozwiązania do tego problemu.

Inteligencja bez reprezentacji wiedzy?

„Klasyczne” podejście do sztucznej inteligencji, oparte na tzw. hipotezie reprezentacji wiedzy, zakłada, że kluczem do inteligencji jest poznanie oparte na wiedzy reprezentowanej wewnątrz. Hipoteza ta pozwoliła naukowcom skoncentrować się na algorytmach uczenia się, planowania, dedukcji itp., jednocześnie w dużej mierze ignorując środowisko, w którym ma się znajdować inteligentny agent. Nawet badania nad wizją komputerową i robotyką były pomyślane jako mające na celu rozwinięcie mechanizmów, które pozwalają inteligentnemu agentowi postrzegać swoje środowisko i działać w nim, ale nadal być tylko danymi wejściowymi i wyjściowymi dla jakiegoś innego centralnego komponentu poznawczego. Ten podstawowy model został zakwestionowany w połowie lat 80. XX wieku. Twierdzono, że interakcja między percepcją a działaniem może tworzyć złożone zachowania i że nie ma potrzeby stosowania komponentu „poznawczego” opartego na wewnętrznej reprezentacji wiedzy, która ma zdolności rozumowania symbolicznego. Poznanie wyłania się z tej interakcji, ale nie jest w nią zaprogramowane. Z tego punktu widzenia „klasyczny” rozkład inteligencji, jaki obserwuje większość badaczy AI, jest przedwczesny. Inteligencja na poziomie człowieka jest zbyt złożona i wiemy o niej zbyt mało, aby móc skutecznie zidentyfikować jej składniki. Zamiast tego badacze podążający za tym poglądem próbują zbudować kompletne systemy, które oddziałują ze swoim środowiskiem, zaczynając od małych, które są tak inteligentne jak, powiedzmy, owady, i budując w górę. Takie podejście rozkłada inteligentne zachowanie zgodnie z jego działaniami, a nie jego funkcjami. Dlatego inteligentny system nie powinien być postrzegany jako składający się z różnych funkcji prowadzących od percepcji do działania poprzez pośrednie zadania poznawcze, takie jak modelowanie i planowanie. Składa się raczej z różnych umiejętności, z których każda przechodzi od percepcji do działania. Na przykład w robocie takie umiejętności mogą obejmować unikanie kolizji, eksplorację i analizowanie widocznej przestrzeni, wyszukiwanie określonych obiektów (takich jak gniazdka elektryczne do ładowania się...) i tak dalej. Rzeczywiście, wiele robotów (w tym komercyjny autonomiczny odkurzacz domowy) zostało wyprodukowanych w ten sposób. Ich poziom inteligencji można porównać do poziomu niektórych owadów (na co nie można kichać!), ale wciąż jest to dalekie od inteligencji na poziomie człowieka. Czas pokaże, czy to oparte na zachowaniu podejście do sztucznej inteligencji rzeczywiście będzie skutecznie zwiększać skalę w przyszłości.

Perspektywy sztucznej inteligencji

Termin „sztuczna inteligencja” został po raz pierwszy ukuty w związku z konferencją w Dartmouth w 1956 roku. Było to spotkanie czołowych badaczy z różnych dziedzin nauki, którzy próbowali ustalić

program badawczy dla raczkującej dziedziny. Ich nastrój był bardzo optymistyczny i spodziewali się wielkich przełomów w ciągu 10 lat. Najważniejszym rezultatem późniejszych badań było uświadomienie sobie, jak trudne były w rzeczywistości problemy. W konsekwencji nieuchronnie ucierpiała reputacja branży.

Jak sobie radziliśmy przez mniej więcej pół wieku, które minęło? Chociaż nie opracowano jeszcze nic zbliżonego do prawdziwej sztucznej inteligencji, zwrócenie się ku bardziej wyspecjalizowanym problemom przyniosło imponujące sukcesy, zarówno pod względem naukowym, jak i komercyjnym. Na przykład dzisiejsze komputery potrafią rozumieć polecenia głosowe, rozpoznawać twarze i obiekty na zdjęciach oraz tworzyć zadowalające tłumaczenia dokumentów technicznych. Jak wspomniano wcześniej, systemy oparte na wiedzy są wykorzystywane w wielu przedsięwzięciach komercyjnych i inżynierskich, często za kulisami. W rzeczywistości wiele rzeczy, o których przywykliśmy czytać w science fiction w gazetach pojawiają się obecnie historie opisujące trwające badania, a czasem nawet istniejące systemy. Obejmują one komputery do noszenia, które są stale połączone z ogólnosięciową siecią informacyjną; rzeczywistość wirtualna i rozszerzona; „inteligentny pył”, czyli liczne małe czujniki, które działają razem jako rozproszony komputer; „inteligentne pokoje”, które za pośrednictwem ekranów wielkości ścian reagują na polecenia głosowe, gesty rąk i mimikę twarzy; i, bardziej ponuro, monitorowanie przez rząd całej komunikacji internetowej. Pomimo tych osiągnięć sztuczna inteligencja jest dziś równie kontrowersyjna, jak na początku. Stało się jasne, że systemy budowane przy użyciu technik AI mogą być bardzo przydatne w praktykowaniu nawet bez osiągnięcia inteligencji na poziomie człowieka. Niektórzy (w tym wielu badaczy sztucznej inteligencji) uważają, że ostateczny cel prawdziwej inteligencji maszynowej jest nieosiągalny. Inni przewidują przyszłość, w której inteligencja maszyn w końcu przewyższy ludzką inteligencję, być może nawet prowadząc do połączenia ludzi i komputerów. Pewien słynny wynalazca i autor przewidział w 1999 r., że do 2029 r. pojawią się bezpośrednie połączenia neuronowe między ludzkim mózgiem a komputerami, a roszczenia maszyn do bycia świadomymi zostaną w dużej mierze zaakceptowane. Przewiduje się, że do 2099 roku nie będzie już wyraźnego rozróżnienia między ludźmi a komputerami. Pozostawiamy czytelnikowi decyzję, czy jest to możliwe, prawdopodobne, a nawet pożądane . . .