ELIZA — jak człowiek może rozmawiać z komputerem

Mgr Małgorzata NALBACH, mgr Krzysztof STUDZIŃSKI

Dzień dobry. Co cię do mnie sprowadza?
Moja mama ciągle na mnie krzyczy —
Opowiedz mi więcej o swojej rodzinie.
Mój tata każe mi się ciągle uczyć —
Kto jeszcze w twojej rodzinie każe ci się ciągle uczyć?
Wszyscy. Mówią, że **jeśli** się nie będę uczyć, to do niczego nie dojdę —
Uzasadnij to.
To oni **tak** mówią —
Dlaczego tak uważasz?
Zadajesz niemądre pytania. Przecież słyszałam —
Nie jestem pewna, czy cię całkowicie rozumiem.
Jesteś taka jak oni —
Jakie związki tu widzisz?
Ty mnie też **nie** rozumiesz. Nie można życia spędzić nad książkami —
Uzasadnij to.
Powtarzasz się **jak** głupia maszyna —
Czy istotnie jest tu jakieś podobieństwo?
Wydaje się, że tak —
Czy jesteś pewna?
Przepraszam za to porównanie —
Usprawiedliwienia nie są potrzebne.
Oni **zawsze** każą przepraszać —
Czy możesz podać konkretny przykład?

Tak rozmawiał pewien człowiek z komputerem. Spróbuj dokładnie prześledzić tok rozmowy; być może zauważysz bardzo ciekawe rzeczy.

Od początku historii komputerów ludzi denerwowały trudności, które musieli pokonywać, żeby maszyna cyfrowa wykonywała ich polecenia. Komputery bowiem są w stanie zrozumieć tylko bardzo precyzyjnie sformułowane polecenia wyrażone w złożonym i sformalizowanym języku programowania. Ograniczało to liczbę ludzi korzystających z komputerów do tylko tych, którzy opanowali pewien język programowania, a więc spełnili wymagania stawiane przez maszynę. Czyli coś, co ma służyć człowiekowi, stawia mu duże wymagania.

Ideałem byłoby, gdyby z komputerem można było porozumieć się tak naturalnie jak podczas wymiany informacji między ludźmi. Tego typu idee ludzie mieli już od dawna. Na pewno znane są powieści fantastyczno-naukowe, gdzie pojawiają się motywy rozmowy z komputerem czy robotem.

Już w roku 1966 Weisenbaum zaprojektował i uruchomił system ELIZA. Nazwa programu pochodzi od imienia głównej bohaterki „Pigmaliona" G. B. Shawa. Założeniem programu ELIZA jest podtrzymywanie dialogu, to znaczy zachowywanie się tak, jak zachowuje się dobrze wychowana osoba z uwagą słuchająca rozmówcy i zachęcająca go do dalszych zwierzeń. Program ten został zaprojektowany w oparciu o metodę słów kluczowych. Maszyna zaprogramowana jest w taki sposób, że z wprowadzanych do niej słów i zwrotów wyszukuje te, które potrafi zidentyfikować i na ich podstawie tworzy odpowiedź. O rodzaju odpowiedzi i temacie rozmowy decyduje zasób słów kluczowych obrany przy projektowaniu programu. Oryginalny program ELIZA prowadził rozmowę w języku angielskim.

W roku 1981 w Instytucie Informatyki Uniwersytetu Warszawskiego powstał analogiczny program pozwalający na rozmowy w języku polskim. Poniżej przedstawimy zasady, w oparciu o które działa taki program.

Podstawową częścią programu ELIZA jest automat złożony z pewnej liczby stanów. Z każdym stanem związane są następujące elementy:
1. nazwa stanu, która pozwala odróżnić go od innych,
2. akcja podejmowana przy wejściu do tego stanu,
3. tekst wypisywany w tym stanie,
4. liczba odpowiedzi w tym stanie. Jeśli liczba odpowiedzi jest równa 0, to stan taki jest stanem

Rozwiązanie zadania o sortowaniu.
Oszacowanie $\frac{n!}{2}$ otrzymujemy zakładając, że do sprawdzenia, czy ciąg jest posortowany, musimy wykonać co najmniej jedno porównanie.
Czas sortowania opisanym algorytmem — więcej niż 75 dób.
Czas sortowania algorytmem 3 — około $6 \cdot 10^{-4}$s.

W 1939 roku A. M. Turing sformułował tezę, że można uważać maszynę za „myślącą", kiedy człowiek, przebywający w innym pokoju niż maszyna i prowadzący z nią rozmowę, nie będzie mógł poznać, czy rozmawia z człowiekiem czy maszyną. ELIZA miała dowieść, że teza ta jest fałszywa: nawet prosty i na pewno bezmyślny program może udawać człowieka wcale sugestywnie. Podobno jeden z prawdziwych („żywych") psychiatrów rozmawiał przez kilkanaście minut z ELIZĄ (za pośrednictwem dalekopisu) przekonany, że rozmawia ze swoją koleżanką po fachu. Powiedział nawet „na przykład żadna maszyna nie potrafiłaby rozmawiać tak, jak Pani". A problem, co to znaczy „myśląca maszyna" nadal jest otwarty.

Rozwiązanie zadania F 181. Zasada zachowania momentu pędu pozwala wyznaczyć prędkość kątową obrotu ω (po odłączeniu napędu) jako funkcję odległości kulki od osi obrotu, mamy bowiem

$$L = (J+Mr^2)\omega = (J+Mr_0^2)\,\omega_0,$$

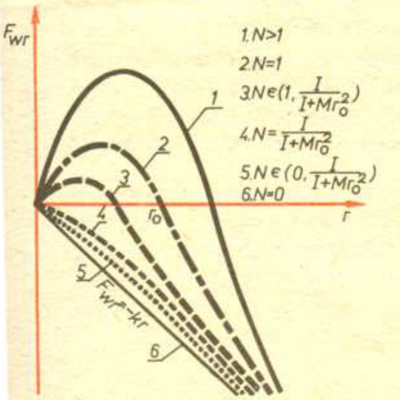
gdzie L — momentu pędu, a J — moment bezwładności wirującej części układu. Działające siły mają postać analogiczną jak w poprzednim zadaniu, składowa siły wypadkowej równoległa do pręta wynosi więc

$$F = (M\omega^2-k)r = \left[M\left(\frac{L}{J+Mr^2}\right)^2-k\right]r.$$

Oznaczając $\Omega = \sqrt{\frac{k}{M}}$ i $\omega_0 = N\Omega$ mamy

$$F = M\Omega^2\left[N^2\left(\frac{J+Mr_0^2}{J+Mr^2}\right)^2-1\right]r.$$

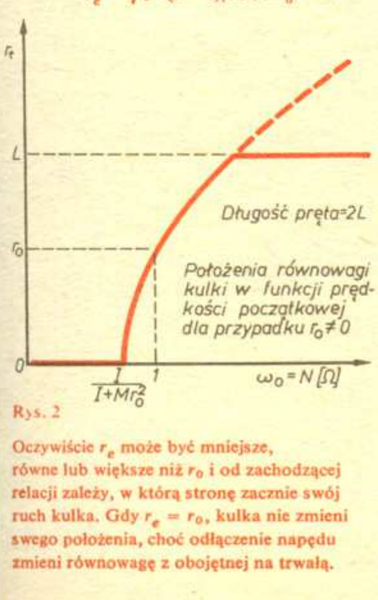
Szkicowe wykresy zależności F od r dla różnych wartości parametru N przedstawia rysunek 1. Od wartości tego parametru zależy położenie miejsc zerowych funkcji (położenia równowagi) i znaki pochodnych w tych punktach (charakter równowagi).



Rys. 1

Po uzyskaniu swobody ruchu kulka wykonuje drgania wokół położenia równowagi trwałej wyznaczonego przez początkową prędkość kątową układu. Położenia równowagi dla $r_0 \neq 0$ przedstawia rysunek 2. Przy małych prędkościach początkowych położenie równowagi r_e znajduje się na osi obrotu, przy większych ω mamy

$$r_e = \sqrt{J\cdot(N-1)/M+Nr_0^2}.$$



Rys. 2

Oczywiście r_e może być mniejsze, równe lub większe niż r_0 i od zachodzącej relacji zależy, w którą stronę zacznie swój ruch kulka. Gdy $r_e = r_0$, kulka nie zmieni swego położenia, choć odłączenie napędu zmieni równowagę z obojętnej na trwałą.

końcowym — na nim kończy się rozmowa. Jeśli liczba odpowiedzi jest równa 1, to z danego stanu następuje bezpośrednie przejście do stanu wskazanego jako następny. Przy liczbie odpowiedzi większej od 1 program próbuje dopasować wyraz wejściowy do jednej z przewidzianych w tym stanie odpowiedzi użytkownika. Kiedy mu się to udaje, następuje przejście do wskazanego dla tej odpowiedzi stanu.

Ponieważ każdy stan jest opisywany w ten sam sposób, cały automat można przedstawić w postaci tabelki, gdzie jeden wiersz opisuje stan. Wiersz taki zawiera następujące informacje:
— nazwa stanu,
— akcje,
— tekst,
— liczba odpowiedzi,
— nazwy stanów docelowych.

Oto fragment automatu, który działa po rozpoznaniu w zdaniu wejściowym jednego ze słów rozumianych przez ELIZĘ jako porównanie, a więc: podobnie, jak, podobny, jednakowy, tacy, taki. Wyrazy odmienne podawane są we wszystkich formach odmiany, w jakich mogą wystąpić.

stan	akcje	tekst	l odp.	odp.	do
sjak	0	—	4	nazw./nazywa./—/.	nazywa/nazywa/enjak/sjak
enjak	2	—	1	—	s
nazywa	0	—	2	—/.	ennazywa/nazywa
ennazywa	1	—	1	—	s

W kolumnie AKCJE występują tu numery akcji, jakie należy wykonać po wejściu do danego stanu. Liczba 0 oznacza, że żadna akcja nie będzie podejmowana. W automacie realizującym ELIZĘ akcje służą do generowania i wypisywania odpowiedzi ELIZY. W innym miejscu pamięci komputera jest przechowywana lista możliwych odpowiedzi. Na każde słowo kluczowe ELIZA może odpowiedzieć na kilka możliwych sposobów. I tak na słowo z podanej już grupy porównań ELIZA podaje, realizując w ten sposób akcję numer 2, jedną z następujących odpowiedzi:
Na czym polega to podobieństwo?
Jakie związki tu widzisz?
Czy istotnie jest tu jakieś podobieństwo?
Na co wskazuje to podobieństwo?

Odpowiedzi te podawane są kolejno. Pozwala to na urozmaicenie dialogu, co powoduje, że ELIZA zbyt często się nie powtarza. Myślnik w kolumnie TEKST oznacza, że żaden tekst nie będzie wypisywany. W tym bowiem przypadku teksty, a więc komentarze lub odpowiedzi programu są generowane, a nie wypisywane bezpośrednio. Gdyby zostały podane w tym miejscu, w każdej sytuacji możliwa by była tylko jedna odpowiedź programu.

Kolumna LODP zawiera liczbę odpowiedzi, a jej znaczenie zostało wcześniej wyjaśnione. W stanach ENJAK oraz ENNAZYWA ELIZA udziela odpowiedzi, po czym przechodzimy do stanu S, w którym jest wczytywane następne zdanie rozmówcy i rozpoczyna się proces rozpoznawania kolejnych wyrazów tego zdania.

W kolumnie ODP umieszczone są oczekiwane w tym stanie wyrazy lub zwroty. Kolejne wyrazy oddzielone są znakiem /. Podawane są tylko tematy wyrazów, końcówki zastępuje kropka. Kropka pasuje do wszystkiego, czyli jeśli jakiś wyraz jest zgodny z podanym tu, na wszystkich literach aż do kropki, przyjmuje się, że jest zgodny do końca. W szczególności sama kropka oznacza dowolny wyraz. I tak, na przykład do wzorca **nazw.** mogą zostać dopasowane następujące wyrazy (o ile wystąpią w zdaniu wejściowym):

nazwisko, nazwiska, nazwa, nazwę itd.

Pozwala to na pominięcie problemów związanych z odmianą wyrazu. Wystąpienie tematu wyrazu jest równoznaczne z wystąpieniem całego wyrazu. Jednocześnie widać, że tak rozumiane tematy wyrazów należy wyodrębniać ze szczególną ostrożnością. Źle wybrany temat może prowadzić do błędnych odpowiedzi.

Każde zdanie rozmówcy ELIZY kończy myślnik. Dopiero po rozpoznaniu końca zdania automat udziela odpowiedzi.

W każdym zdaniu może wystąpić 0, 1 lub więcej słów kluczowych. Jeśli nie ma żadnego słowa kluczowego, zdanie jest dla ELIZY niezrozumiałe i podawana jest jedna ze standardowych dla tej sytuacji odpowiedzi:

Nie jestem pewna, czy cię całkowicie rozumiem.
Proszę mów dalej.
Co twoim zdaniem z tego wynika.
Czy cię denerwuje mówienie o tych sprawach?

W przypadku, gdy w zdaniu wystąpi kilka słów kluczowych, jest wybierane słowo uznawane za najistotniejsze, czyli to, które ma najwyższy priorytet. W każdym stanie słowa kluczowe w kolumnie ODP uporządkowane są według priorytetów, od największego do najmniejszego. Ponadto w stanie, który został osiągnięty po rozpoznaniu jednego ze słów kluczowych, sprawdzane jest, czy dalej w zdaniu nie wystąpiło przypadkiem słowo kluczowe o wyższym od tego priorytecie i jeśli tak, następuje przejście do stanu odpowiadającego wystąpieniu tego słowa. W naszym przykładzie widać, że słowa kluczowe oznaczające porównanie mają bardzo wysoki priorytet. Wyższy od nich mają tylko słowa oznaczające nazwy lub nazwiska. W podanym na początku przykładzie dialogu zostały wyróżnione słowa kluczowe, na które ELIZA reagowała udzielając stosownych odpowiedzi.

Można również zaobserwować jeszcze jedną cechę ELIZY. Otóż wszystkie wyrazy bliskoznaczne traktuje ona w ten sam sposób. Bez względu na to czy w zdaniu wystąpi wyraz jak, jednakowy, czy taki, odpowiedź będzie taka sama.

Kolumna DO zawiera nazwy stanów, do których następuje przejście po rozpoznaniu kolejnego słowa ze zdania użytkownika. W kolumnie DO musi być tyle samo elementów co w kolumnie ODP. Jeśli zostanie rozpoznana pierwsza z listy odpowiedzi w kolumnie ODP, następuje przejście do pierwszego stanu z listy stanów w kolumnie DO, przy drugiej do drugiego, itd. W naszym przykładzie po rozpoznaniu wzorca **nazw.** lub **nazywa.** następuje przejście do stanu NAZYWA, po rozpoznaniu końca zdania (znak –) do stanu ENJAK, w pozostałych przypadkach sterowanie pozostaje w stanie SJAK i jest sprawdzany następny wyraz ze zdania użytkownika.

Na przykładzie programu ELIZA widać, że duże efekty można czasem osiągnąć bardzo prostymi środkami. Omawiany program konwersacyjny nie zmusza użytkownika do korzystania z ograniczonego podzbioru języka, który przyjęto określając strukturę programu, ale raczej nakłania go do zachowania takiej swobody wypowiedzi, z jakiej można korzystać w codziennej rozmowie. ELIZĘ zaprogramowano w taki sposób, aby ukryć przed użytkownikiem przypadki, w których program go nie rozumie.

Czy algorytmy podawane w szkole są poprawne?

Przypuśćmy, że w zadaniu z fizyki potrzebne jest wyliczenie wartości

$$y_1 = \frac{1}{x_1}, \quad y_2 = \frac{1}{x_2},$$

gdzie x_1 i x_2 są pierwiastkami równania kwadratowego

$$x^2 - 2\cdot 10^5 x + 2 = 0.$$

Rozwiązanię wydaje się proste: współczynniki mają niewiele cyfr znaczących i można przypuszczać, że kalkulator liczący z dokładnością do 8 cyfr znaczących powinien wystarczyć. Spróbujmy:

$$\Delta = (2\cdot 10^5)^2 - 4\cdot 2 \approx 4\cdot 10^{10}, \quad \sqrt{\Delta} \approx 2\cdot 10^5,$$

$$x_1 \approx \frac{2\cdot 10^5 - 2\cdot 10^5}{2} = 0, \qquad x_2 \approx \frac{2\cdot 10^5 + 2\cdot 10^5}{2} = 2\cdot 10^5,$$

$$y_1 \text{ nie istnieje}, \qquad y_2 = 5\cdot 10^{-6}.$$

W rzeczywistości

$$x_1 \approx 10^{-5}, \qquad x_2 \approx 2\cdot 10^5 - 10^{-5},$$

$$y_1 \approx 10^5, \qquad y_2 \approx 5\cdot 10^{-6}(1 + 5\cdot 10^{-11}),$$

a więc można uznać, że x_2 i y_2 zostały istotnie obliczone poprawnie.

Jak jednak uzasadnić przyczynę tak dużego błędu x_1 i y_1?
W jaki sposób unikać takich błędów? Odpowiedź w numerze.

Rozwiązanie zadania M 410. Oznaczmy pole tablicy leżące na przecięciu i-tego wiersza i j-tej kolumny przez (i, j). Łatwo zauważyć, że zamiana miejscami dowolnych dwóch wierszy lub dwóch kolumn nie zmienia rozważanych własności tablicy. Weźmy największe k, dla którego można tak poprzestawiać wiersze i kolumny tablicy, aby na polach (i, i) dla $i \leqslant k$ znajdowały się zera. Niech a_{ij} będzie liczbą znajdującą się na polu (i, j) po tym przestawieniu. Zauważmy, że $a_{ij} > 0$ dla $i, j > k$ (w przeciwnym wypadku przestawiając wiersze i-ty i $k+1$-szy i kolumny j-tą i $k+1$-szą otrzymalibyśmy zera na miejscach (i, i) dla $i \leqslant k+1$). Jeśli $a_{lm} = 0$ dla pewnych $l \leqslant k$ i $m > k$, to $a_{jl} > 0$ dla wszystkich $j > k$ (inaczej, przestawiając kolumny l-tą i m-tą otrzymalibyśmy zero na miejscu (j, m), $j, m > k$). Stąd

$$\sum_{j=k+1}^{n} (a_{ij} + a_{ji}) \geqslant n-k \text{ dla } i \leqslant k$$

i $\displaystyle\sum_{i,j=k+1}^{n} a_{ij} \geqslant (n-k)^2.$

Ponieważ $a_{ii} = 0$ dla $i \leqslant k$ to

$\displaystyle\sum_{j=1}^{n} (a_{ij} + + a_{ji}) \geqslant n$ dla $i \leqslant k$. Mamy więc

$$\sum_{i,j=1}^{n} a_{ij} = \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} (a_{ij}+a_{ji}) + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=k+1}^{n} (a_{ij}+a_{ji}) + \sum_{i,j=k+1}^{n} a_{ij} \geqslant$$

$$\geqslant \frac{1}{2} k\cdot n + \frac{1}{2} k(n-k) + (n-k)^2 = \frac{1}{2}(n^2 + (n-k)^2) \geqslant \frac{1}{2} n^2$$

(w sumie $\displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n} (a_{ij}+a_{ji})$ liczba a_{ij} dla $i, j \leqslant k$ występuje dwa razy i raz dla $i > k$ lub $j > k$).

Rozwiązanie zadania M 411. Weźmy dowolną prostą l na płaszczyźnie. Rzut prostopadły W_i na l jest odcinkiem. Niech A_i będzie jego „lewym" końcem, a B_i „prawym" (ustalamy pewien zwrot na l). Niech A_{i0} będzie najdalej „na prawo" położonym punktem spośród A_i. Punkt A_{i0} należy do każdego odcinka A_iB_i (W_{i0} przecina W_i), a więc prosta k prostopadła do l i przechodząca przez A_{i0} przecina wszystkie wielokąty $W_1, W_2, \ldots, W_n$.