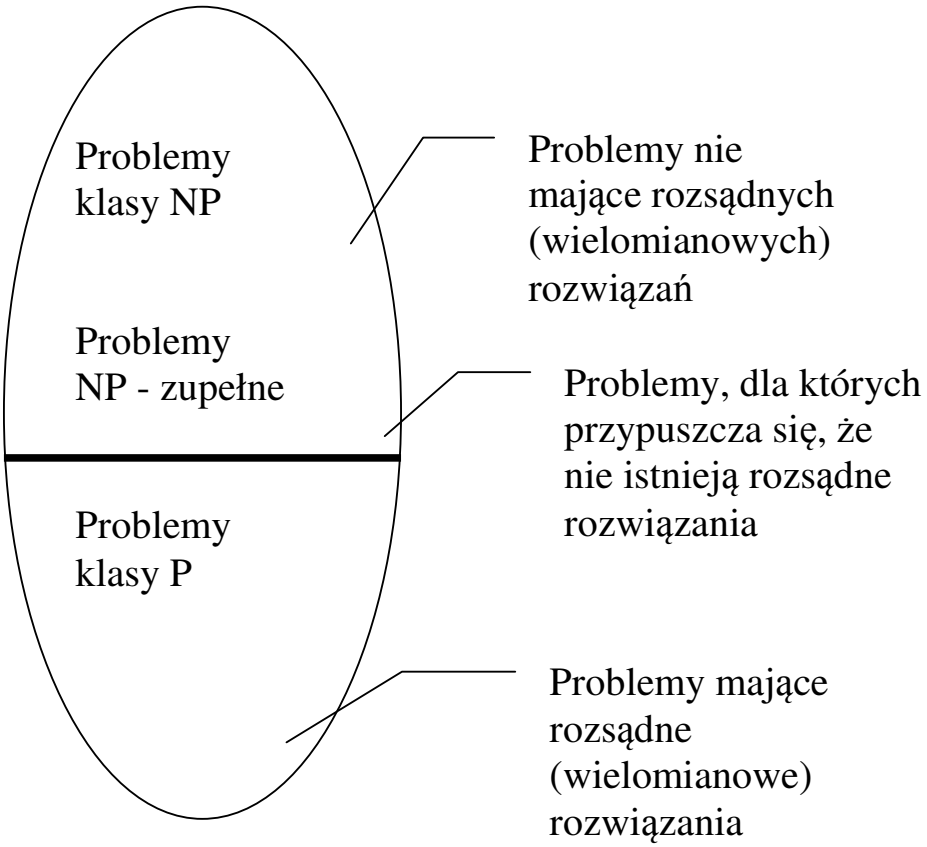
Temat: Klasa problemów NP i NP- zupełnych.
Nieobliczalność i nierozstrzygalność.



1. Przykłady problemów klasy NP

Wszystkie przykłady problemów klasy NP przedstawimy w wariancie decyzyjnym, czyli takim, w którym poszukujemy odpowiedzi na pytanie: "Czy istnieje obiekt opisany w zadaniu?”

a) Generowanie cyklu Hamiltona w spójnym grafie

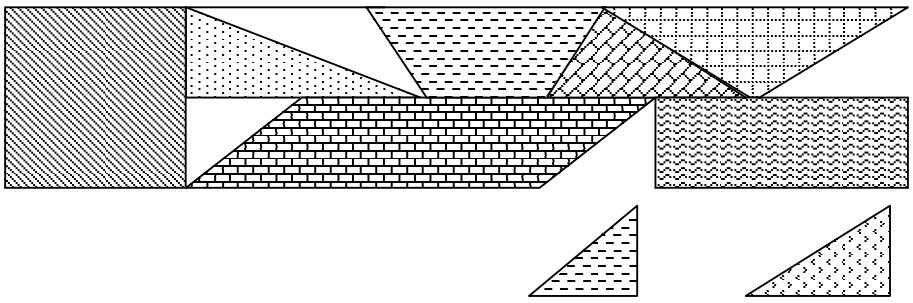
WP: Graf spójny, niezorientowany i *s* - wybrany wierzchołek grafu

WK: Sprawdzenie, czy w grafie istnieje cykl zawierający wszystkie wierzchołki dokładnie raz, zaczynający się w wierzchołku *s*.

b) Problemy ułożeń dwuwymiarowych

WP: *n* wielokątów

WK: Czy z danych wielokątów da się ułożyć prostokąt ?

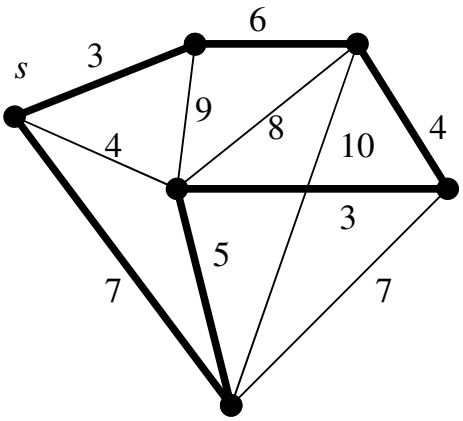


c) Problem komiwojażera

WP: Graf spójny, niezorientowany, z wagami,
wierzchołek startowy *s* i liczba *K*.

WK: Czy istnieje w grafie cykl rozpoczynający się w wierzchołku *s*, zawierający wszystkie wierzchołki (co najmniej raz), o całkowitym koszcie nie większym niż *K*.

Przykład 1



Odpowiedź "tak" dla *K*=30
Zaznaczony cykl jest potwierdzeniem.
Odpowiedź "nie" dla *K* = 27.

d) Problem ułożenia planu

WP: Podane są określone godziny, w których każdy z *n* nauczycieli jest do dyspozycji oraz określone godziny zajęć, w których można przeprowadzić każdą z *m* lekcji. Dodatkowo są podane liczby godzin zajęć, które ma przeprowadzić każdy nauczyciel z każdą klasą.

WK: Należy stwierdzić, czy możliwe jest takie dopasowanie nauczycieli, klas i godzin, aby wszystkie ograniczenia były uwzględnione, tzn. aby żadni dwaj nauczyciele nie uczyli tej samej klasy w tym samym czasie i aby żadne dwie klasy nie miały lekcji z tym samym nauczycielem w tym samym czasie.

e) Ustalanie prawdy logicznej (problem spełnialności)

WP: Zdanie z rachunku zdań, zawierające podstawowe asercje: $\wedge, \vee, \neg, \Rightarrow$ i formuły atomowe

WK: Sprawdzenie, czy istnieje takie wartościowanie formuł atomowych, dla którego formuła jest prawdziwa.

Przykład 2

Dla formuły $\neg(E \Rightarrow F) \wedge (F \vee (D \Rightarrow \neg E))$ takie wartościowanie istnieje i jest określone następująco:

$E = 1,\ D = 0,\ F = 0$

Dla formuły $\neg((D \wedge E) \Rightarrow F) \wedge (F \vee (D \Rightarrow \neg E))$ takie wartościowanie nie istnieje.

Rozmiarem zadania problemu spełnialności jest *n* liczba asercji w danej formule.

f) Kolorowanie grafu

WP: Graf niezorientowany i spójny oraz liczba barw *c*.

WK: Należy stwierdzić, czy można przyporządkować wszystkim wierzchołkom grafu barwy określone numerami od 1 do *c* tak, aby dwa incydentne wierzchołki nie były etykietowane tym samym kolorem.

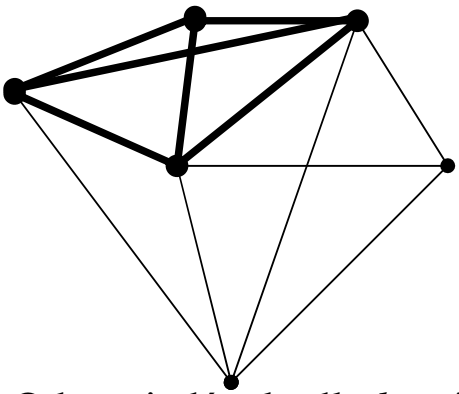
g) Problem kliki

WP: Graf spójny i niezorientowany *G* oraz liczba naturalna *k*.

WK: Czy graf zawiera klikę (podgraf pełny) o rozmiarze *k*.

Graf pełny, to graf, w którym każdy wierzchołek incyduje z wszystkimi pozostałymi.

Przykład 3



Odpowiedź tak, dla $k = 4$. Odpowiedź "nie" dla $k=5$.

h) Sprawiedliwy podział

WP: Skończony zbiór obiektów S, którym przyporządkowano indeksy od 1 do n.

WK: Czy można dany zbiór obiektów podzielić na dwa rozłączne podzbiory S_1 i S_2 tak, aby miały taką samą wartość całkowitą. Wartość całkowita podzbioru to suma indeksów jego elementów.

Przykład 4
Dla $n = 4$, $S = \{1, 2, 3, 4\}$
Odpowiedź "tak": $S_1 = \{1, 4\}$ i $S_2 = \{2, 3\}$

Dla $n = 5$, $S = \{1, 2, 3, 4, 5\}$
Odpowiedź ?: Sprawdź

i) Szeregowanie zadań niezależnych

WP: Program złożony ze skończonej liczby zadań; liczby naturalne d_t oznaczające czas realizacji tych zadań oraz liczba naturalna d oznaczająca czas wykonania całości zadania. Zakładamy, że zadania mogą być realizowane w dowolnym porządku, ale każde z zadań musi być wykonane w całości na jednym procesorze.

WK: Czy program ten może być zrealizowany w żądanym czasie przy dwóch identycznych procesorów.

Przykład 5

Zadania: $a,\ b,\ c,\ d$

Czasy realizacji zadań d_i: 1, 2, 3, 4

Czas wykonania całości zadania d: 5

Odpowiedź tak: Procesor nr 1: a, d; Procesor nr 2: b, c

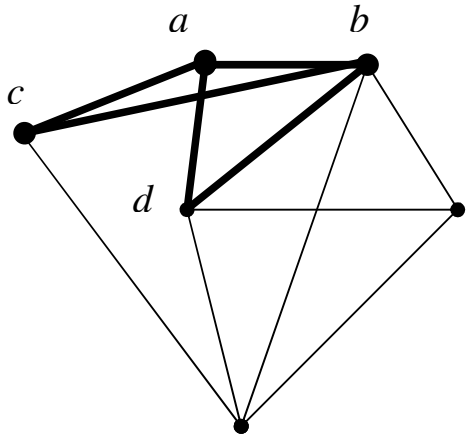
j) Izomorfizm z podgrafem

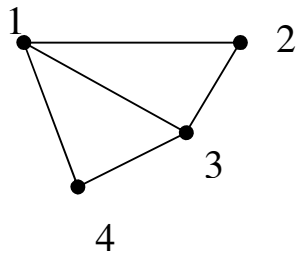
WP: Grafy G i H

WK: Czy graf G zawiera podgraf izomorficzny z H.

Przykład 6

Graf G Graf H





Odpowiedź "tak": Zaznaczony w grafie G podgraf, zawierający wierzchołki a, b, c, d jest izomorficzny z grafem H;
izomorfizm: a – 1, b – 3, c – 4, d - 2

2. Klasa NP - zupełnych

Def. Mówimy, że problem decyzyjny *A* jest przekształcalny w problem decyzyjny *B*, jeśli istnieje przekształcenie $f: A \rightarrow B$ o następujących własnościach:
- odpowiedzi do problemów *A* i *B* są zgodne,
- wartość *f*(*A*) można obliczyć w czasie wielomianowo zależnym od rozmiaru zadania *A*.

Lemat o przekształcalności
Jeżeli problem *A* jest przekształcalny w problem *B* oraz *A* jest problemem klasy NP, to także problem *B* jest problemem klasy NP.

Momentem przełomowym w historii badań nad problemami klasy NP był rok 1971. Stephen Cook pokazał wówczas, że klasa NP zawiera problemy uniwersalne, tzn. takie, iż każdy problem klasy NP jest w nie przekształcalny. Problemy klasy o tej własności nazywamy **NP – zupełnymi**.

Tw. Cooka – Karpa (1971 – 72)
Wszystkie problemy od a) do j) w punkcie 1 są NP- zupełne.

Lemat o przechodniości
Jeżeli problem *A* jest NP – zupełny i jest przekształcalny w problem *B* oraz *B* jest problem klasy NP, to *B* jest NP – zupełny.

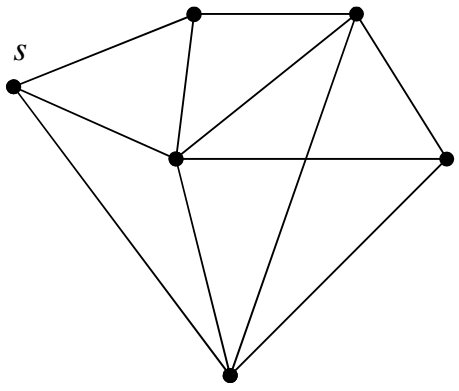
Przykład 7

Zauważmy, że problem cyklu Hamiltona można łatwo przekształcić w problem komiwojażera. W tym celu zastępujemy badany graf G o n wierzchołkach grafem pełnym o tym samym zbiorze wierzchołków. Odległości określamy wzorem:

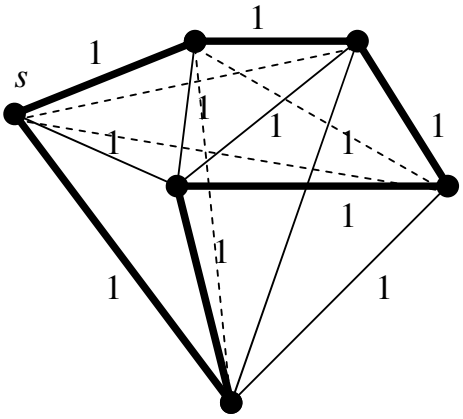
$$d(i,j)=\begin{cases}1 \text{ jeśli } i \text{ oraz } j \text{ tworzą krawędź grafu}\\ 2 \text{ w przeciwnym przypadku}\end{cases}$$

Drogą komiwojażera będziemy nazywali odtąd dowolną (tzn. niekoniecznie najkrótszą) drogę zamkniętą przechodzącą dokładnie raz przez wszystkie wierzchołki. Składa się ona z n krawędzi, więc nigdy nie będzie krótsza niż n. Jej długość wyniesie dokładnie n wtedy, gdy komiwojażer będzie się stale poruszał wzdłuż krawędzi grafu G. Innymi słowy, gdy G będzie grafem Hamiltona. Zatem, aby zbadać hamiltonowskość grafu G, wystarczy rozwiązać problem komiwojażera dla macierzy odległości określonej powyższym wzorem i z ograniczeniem $K=n$.

Przykład 7



Problem cyklu Hamiltona



przekształcony do problemu komiwojażera dla K=6

Przykład 8

Nazwijmy obiekty, o których mowa w problemie sprawiedliwego podziału, zadaniami. Indeks obiektu-zadania przyjmijmy za czas realizacji tegoż zadania. Przyjmijmy za d połowę łącznej wartości wszystkich tych obiektów. Przekształciliśmy w ten sposób problem sprawiedliwego podziału w szeregowanie zadań niezależnych.

3. Hipoteza P≠NP

Każdy problem z klasy NP-zupełnych daje się rozwiązać strategią powrotów. Jednak w najgorszym przypadku potrzebny jest czas wykładniczy, gdy odpowiedź na pytanie zawarte w warunku końcowym problemu wymaga systematycznego zanalizowania wszystkich możliwych rozszerzeń rozwiązania częściowego.

Z drugiej jednak strony, dla wszystkich problemów NP-zupełnych charakterystyczne jest to, że gdybyśmy mieli do dyspozycji wyrocznię, która dla konkretnych danych udzieli nam odpowiedzi "tak" na pytanie zawarte w warunku końcowym problemu i potwierdzi swoją odpowiedź wskazując rozwiązanie, to weryfikacja tego wskazanego rozwiązania może być zrealizowana algorytmem wielomianowym.

Na przykład, gdyby wyrocznia wskazała nam cykl Hamiltona w grafie, to sprawdzenie, czy rzeczywiście jest to cykl Hamiltona może być zrealizowane w czasie O(n). Podobnie, w przypadku problemu spełnialności. Gdybyśmy znali wartościowanie formuł atomowych, przy którym sprawdzana formuła jest prawdziwa, to weryfikację tej prawdziwości też można zrealizować algorytmem wielomianowym.

Wspomniana wyrocznia to **algorytm niedeterministyczny,** który potrafi dokonywać poprawnego wyboru w każdej fazie realizacji strategii powrotów. W trakcie wybierania spośród możliwych rozszerzeń rozwiązania częściowego algorytm niedeterministyczny rozważa tylko ten wariant, który prowadzi do poprawnego rozwiązania całkowitego. Nie nakładamy żadnych warunków na skuteczność działania tego algorytmu w przypadku rozwiązań z odpowiedzią "nie". Algorytm niedeterministyczny może więc czasem nie udzielić odpowiedzi, jednak obecność procedury weryfikacyjnej gwarantuje, że nigdy nie udzieli on odpowiedzi błędnej. Mówimy, że ma on złożoność wielomianową, gdy taką złożoność ma procedura weryfikacji.

Czas na uzasadnienie nazw klas:

- **NP z ang. nondeterministic polynomial - wielomianowy niedeterministycznie**

- **NPC (NP- zupełne) z ang. nondeterministic polynomial complete - wielomianowy niedeterministycznie, zupełny**

- **P z ang. Polynomial** - klasa algorytmów wielomianowych

Hipoteza P≠NP

Następujące warunki są równoważne:

- **Istnieje problem NP- zupełny, który jest trudno rozwiązalny.**

- **Każdy problem NP - zupełny jest trudno rozwiązalny.**

Jeżeli ta hipoteza jest prawdziwa, to nie warto szukać rozsądnych rozwiązań dla problemów NP - zupełnych, bo ich po prostu nie ma.

Hipoteza P=NP

Następujące warunki są równoważne:

- **Istnieje problem NP- zupełny, który jest łatwo rozwiązalny (należy do klasy P).**

- **Każdy problem NP - zupełny może być rozwiązany w czasie wielomianowym.**

Jeżeli ta hipoteza jest prawdziwa, to jeżeli uda się podać rozwiązanie wielomianowe jednego problemu NP- zupełnego, to tym samym rozwiązalne wielomianowo będą pozostałe problemy z tej klasy.

4. Nieobliczalność i nierozstrzygalność

"*Wprowadź do komputera właściwe oprogramowanie a otrzymasz, co tylko zechcesz. Twoje chęci może ograniczać jedynie sama maszyna - oprogramowanie nie stawia żadnych ograniczeń*".

Times, kwiecień 1984

Istnieją niestety problemy algorytmiczne, dla których nie ma żadnego algorytmu w sensie programu w konwencjonalnym języku programowania. Można nawet ten fakt braku rozwiązania potwierdzić dowodem.

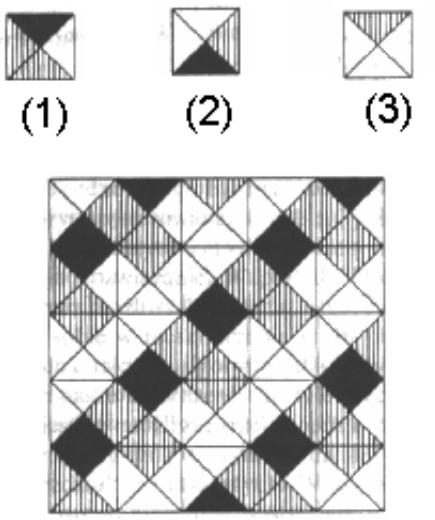
Def. 5 Problem algorytmiczny, dla którego nie istnieje algorytm go rozwiązujący nazywamy **nieobliczalnym**, a jeżeli problem jest decyzyjny to **nierozstrzygalnym**.

a) Problem domina

WP: Skończony zbiór *T* opisów kart. Zakłada się, że dostępna jest nieskończona liczba kart każdego opisu.

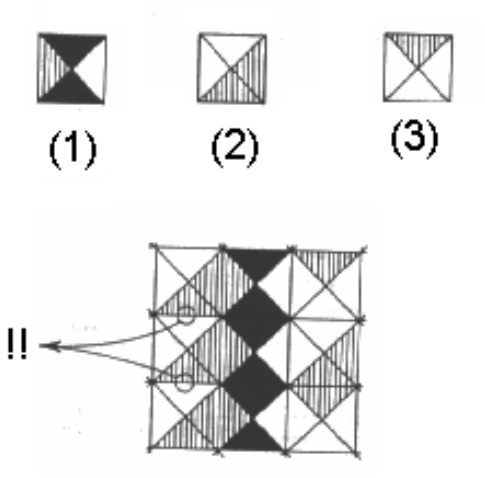
WK: Odpowiedź na pytanie: Czy dowolną skończoną powierzchnię o dowolnym rozmiarze można pokryć za pomocą kart, których opis znajduje się w zbiorze *T*, w ten sposób, aby wzory dwóch przylegających powierzchni były identyczne.

Przykład 9



rys. 1

Można pokazać, że przy zestawie kart z rys. 1 da się ułożyć dowolną powierzchnię wzorem takim jak na powierzchni przykładowej 5 na 5.



rys. 2

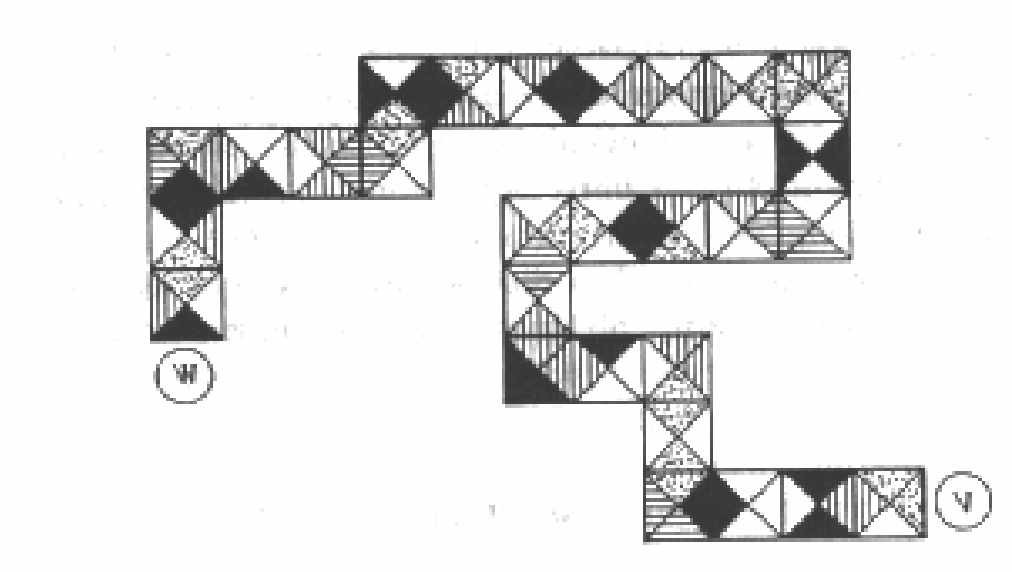
Po zamianie dolnych kolorów wzorów (1) i (2), okazuje się, co można całkiem prosto udowodnić, że takimi elementami nie da się ułożyć nawet bardzo małych powierzchni. Rys. 2 jest ilustracją tego faktu.

Zasada, że nieograniczoność liczby przypadków, które trzeba sprawdzić, pociąga za sobą nieobliczalność (czy też nierozstrzygalność) jest błędna. Zmodyfikujmy problem domina:

WP: Skończony zbiór *T* opisów kart. Zakłada się, że dostępna jest nieskończona liczba kart każdego opisu. Dane są również współrzędne dwóch punktów *V* i *W* na nieskończonej siatce liczb całkowitych, które są odpowiednio początkiem i końcem domina.

WK: Sprawdzić, czy można połączyć punkty *V* i *W* "wężem" domina składającym się z kart o wzorach ze zbioru *T* tak, aby każde dwa stykające się krawędzie były tego samego koloru.

Przykład 10



Również i ta odmiana problemu domina jest nierozstrzygalna.

b) odpowiedniość słów

WP: Dane są dwa różne ciągi n - elementowe słów skonstruowanych nad pewnym alfabetem.
Ciągi: $X = (X_1, X_2, ..., X_n)$, $Y = (Y_1, Y_2, ..., Y_n)$

WK: Odpowiedź na pytanie: Czy możliwe jest wybranie pewnego ciągu słów z ciągu X, których złożenie da nowe słowo, w taki sposób, aby w wyniku tego samego ciągu wyborów, ale słów z ciągu Y, powstało to samo słowo.

Przykład 11

	1	2	3	4	5
X	*abb*	*a*	*bab*	*baba*	*aba*
Y	*bbab*	*aa*	*ab*	*aa*	*a*

Jest odpowiedniość: 2, 1, 1, 4, 1, 5

	1	2	3	4	5
X	*bb*	*a*	*bab*	*baba*	*aba*
Y	*bab*	*aa*	*ab*	*aa*	*a*

Nie ma odpowiedniości

Nieograniczona istota problemu odpowiedniości słów wynika z faktu, że długość ciągu indeksów, które mogłyby dać wspólne słowo, nie jest ograniczona żadną obliczalną funkcją. Interesujące jest to, że wystarczy zrezygnować z założenia, że ciągi indeksów dające wspólne słowo muszą być takie same, a problem staje się rozstrzygalny. W drugim przykładzie ciąg słów z *X*: 3, 2, 2 i ciąg słów z *Y*: 1, 2 daje to samo słowo.

c) problem stopu

Rozważmy następujący algorytm A_1:
while (x!=1) x=x-2;

Zakładając, że poprawne dane wejściowe dla tego algorytmu składają się z liczb naturalnych, jest oczywiste, że algorytm A_1 zatrzymuje się tylko dla liczb nieparzystych. Nie zatrzymuje się dla liczb parzystych, są one bowiem stale zmniejszane o 2. Liczba 1 jest więc "ominięta", a program wykonuje się w nieskończoność dla liczb 0, -2, -4, -6, itd. Zatem dla tego szczególnego algorytmu stwierdzenie, czy jakieś dane

spowodują jego zatrzymanie, jest bardzo proste: należy jedynie sprawdzić, czy liczba wejściowa jest parzysta czy nieparzysta i dać właściwą odpowiedź.

Rozważmy teraz drugi, podobny algorytm A_2:

```
while (x!=1)
    if (x % 2==0) x=x/2;
    else x=3*x+1;
```

Algorytm A_2 zawsze się zatrzymuje - tak wskazują wszystkie testy empiryczne. Natomiast nikt jak dotąd nie potrafi udowodnić tego faktu.

Podany przykład jest zapowiedzią trudności związanych z algorytmizacją metod dowodzenia problemu stopu.

WP: Tekst poprawnego częściowo programu *R* w języku *L* oraz dane *X* spełniające warunek początkowy problemu, który rozwiązuje program *R*.

WK: Odpowiedź na pytanie: Czy program *R* zatrzyma się po wykonaniu go dla danych *X* ?

Problem nie jest sztuczny, choć może się tak wydawać na pierwszy rzut oka. Można przecież uruchomić program *R* dla danych *X* i jeżeli zatrzyma się to znaczy, że się zatrzymuje.
W przypadku jednak, gdy program się nie zatrzymuje musimy podjąć subiektywną decyzję o jego przerwaniu, ale do końca nie będziemy pewni, czy program nie zatrzymałby się, gdybyśmy jeszcze trochę poczekali. Trudność więc polega na zdecydowaniu, kiedy zakończyć czekanie i dać odpowiedź "nie".