

Synteza mowy
Pracownia 2
Zajęcia 4 i 5

Zadanie 1. (2p) Zaprezentuj (być może zmodyfikowany) program do wymowy z pierwszej listy (zadanie 1). Na KNO pojawi się zbiór do testowania tego programu, wszystkie testy powinny przejść bezbłędnie (kontrolę musisz samodzielnie zaimplementować w swoim kodzie).

Zadanie 2. (5p) Na KNO pojawi się podział CMU-Dicta na część uczącą i testową. Zaproponuj dowolny algorytm, który rozwiązuje problem sylabizacji fonetycznej reprezentacji danego słowa angielskiego (a ogólniej – słowa z CMU-Dicta). Algorytm powinien:

- spamiętywać ciągi spółgłosek w kontekstach samogłoskowych lub bez,
- wykorzystywać obie te informacje do podjęcia decyzji o podziale,
- radzić sobie (jakoś) również w sytuacji, gdy danego ciągu nie ma w danych uczących.

Jak wiele błędów popełnia Twój program na zbiorze walidacyjnym? Uwaga: to zadanie będzie miało kontynuację.

Zadanie 3. (5+2p) Zmodyfikuj program grający melodię by korzystał z innych instrumentów (znajdziesz je na KNO). Niestety, instrumenty te nie mają wszystkich tonów, brakujące dźwięki powinieneś intepolować, poniżej wyjaśniamy, jak to uczynić:

a) W muzyce mamy następującą sekwencję dźwięków:

c1 c#1 d1 d#1 e1 f1 f#1 g1 g#1 a1 a# h c2 c#2 d2 ...

b) Każdy kolejny dźwięk ma częstotliwość większą od poprzedniego $\alpha = \sqrt[12]{2}$ razy

c) Zatem jeżeli w bazie mamy dźwięki c1,e1,g1,c2,e2,... a do zagrania mamy dźwięk d1, wówczas możemy albo otrzymać go „zagęszczając” α^2 krotnie dźwięk c1, albo rozrzedzając w ten sam sposób dźwięk e1.

d) W powyższym przykładzie wybór nie był oczywisty, generalnie natomiast wybieramy ten dźwięk, który jest bliżej, czyli dla f1 wybralibyśmy e1 a nie g1.

e) Interpolacje powinna być liniowa. Oczywiście zmienia ona długość dźwięku, ale nie musisz się tym przejmować (i tak za długie dźwięki trzeba skracać, a za krótkie wydłużać ciszą).

Wystarczy, że zagrasz jakąkolwiek melodię (może być z pierwszej listy), dodatkowy bonus (niewliczający się do maksimum) otrzymasz za utwór, w którym jednocześnie gra więcej instrumentów, każdy coś trochę innego. Przykładowy utwór będzie na KNO. **Zadanie to nie wlicza się do maksimum.**

Zadanie 4. (7+1p) Napisz program, który dzieli plik dźwiękowy na części, o nazwach pobranych z pliku tekstowego. Podział powinien być po ciszach (to znaczy punkt podziału powinien wypaść w cichym momencie), przy czym interesują nas następujące właściwości tego podziału:

1. Części ma być dokładnie tyle, co nazw plików.
2. Części powinny być pozbawione ciszy z początku i końca.
3. W miarę możliwości nie powinny mieć ciszy w środku.

Twój program powinien dobrze podzielić któryś z plików z nagraniem sylabami. Dodatkowa premie, jeżeli podzieli również jakiś plik nieodszumiony.

Zadanie 5. (6+2p) Napisz syntezytor sylabiczny, czyli taki, który syntetyzuje polski tekst korzystając z nagranych sylab. W przypadku braku sylaby powinien wybrać jedną z następujących strategii:

1. Połączyć z nagraniami pojedynczych fonemów,
2. Znaleźć dwie sylaby w zamian. Na przykład zamiast **popr** wykonać nagranie z „zająknięciem”, czyli **po-opr** (możesz też trochę skrócić oba o)
3. znaleźć sylabę podobną, dopuszczając wszakże jedynie bardzo drobne rozbieżności.

Akcent powinien być zrealizować podgłaszając akcentowaną sylabę (ewentualnie ścisząc nieakcentowaną).

Bonus zależy od tego, czy (i jak ładnie) zrealizujesz szukanie sylaby podobnej (1 – jakkolwiek, ale raczej sensownie, 2 – bardzo fajnie). Zadanie z syntezytorem sylabicznym będzie miało dalszy ciąg i można je będzie oddawać bez straty podczas pierwszych zajęć Pracowni 3.

Zadanie 6. (3p) Za pomocą dowolnego algorytmu lub ręcznie wygeneruj plik tekstowy, w którym mamy wszystkie 300 sylab z Pierwszej Porcji Polskich Sylab i dla każdej z nich jest podany tekst angielski, który zmusi program festival do wymówienia tej sylaby. Format powinien być następujący:

j-a-k yak
si-e shy
cz-u choo
j-e-sz yesh
dzi-i jee
si-a-j shyay

Wyślij wynik na KNO. Zadanie będzie miało dalszy ciąg.