

**Sztuczna inteligencja**  
**Pracownia 5**  
**Część 2**  
**Inne Autka**

Wszystkie zadania są z gwiazdką!

**Zadanie 1. (4p)** W zadaniu tym rozważamy inny niż na poprzedniej liście, deterministyczny mechanizm działania samochodu. Stanem samochodu jest również położenie  $(x,y)$  oraz prędkość  $(v, \alpha)$ , gdzie  $\alpha$  jest kątem między osią X a wektorem prędkości. Prędkość i kąt są dyskretne (w jednym ruchu można zmienić każdą z nich o co najwyżej 1), ale położenie wyrażane jest liczbami rzeczywistymi, co sprawia, że wyliczenie stanów przestaje wchodzić w grę. Mamy następujące akcje:

- . - toczenie się
- r. - skręt w prawo
- l. - skręt w lewo
- a. - przyśpieszenie
- ar. - przyśpieszenie i skręt w prawo
- al. - przyśpieszenie i skręt w lewo
- b. - hamowanie
- br. - hamowanie i skręt w prawo
- bl. - hamowanie i skręt w lewo

które, jak widać, można sklejać w 1 napis i dzielić ze względu na kropki. Ponieważ model jest deterministyczny, a obliczenia mogą być długotrwałe, oddając zadanie należy:

- a) zaprezentować ciąg akcji (w pliku `actions_for_<task-name>`), który doprowadza do mety dla każdego przypadku testowego (będzie udostępniony symulator, który sprawdzi poprawność ścieżki i jej długość)
- b) umieścić wszystkie pliki z akcjami i torami w jednej kartotece, uruchomić plik `test.sh`.
- c) być gotowym, jeżeli prowadzący poprosi, o powtórzenie części obliczeń

Dokładny opis mechaniki samochodu znajduje się w pliku `char_model2.py`.

**Zadanie 2. (3p+X)** W tym zadaniu rozważamy inny model autka (`char_model3.py`), w którym położenie zmienia się w nieco bardziej realistyczny sposób (choć autko sterowane jest tymi samymi poleceniami). Najważniejsze różnice:

- a) Zarówno kierunek, jak i kąt są również zmiennymi rzeczywistymi,
- b) uzględnione są opory, związane z tarcie (składowa stała) oraz z oporami powietrza (zależnymi od prędkości i kwadratu prędkości)
- c) to, o ile skęcimy wykonując akcję l lub r zależy od prędkości
- d) Akcje mają b.dużą rozdzielczość czasową: na przykład do rozpędzenia autka do prędkości rzędu pół piksela na cykl potrzebne jest kilkadziesiąt cykli przyśpieszania.

Jest też jeszcze jedna różnica: zadanie tymczasowo nie ma rozwiązania wzorcowego. Jest wyznaczona ręcznie trasa dla `task2.txt`, która jest mocno nieoptymalna (kilka razy autko prawie się zatrzymuje, żeby zmieścić się w zakręcie), a ma długość 1800 akcji.

Warunkiem zaliczenia zadania jest napisanie programu, który działa dla różnych torów, a dla toru z `task2.txt` generuje co najwyżej 1700 akcji. Za każdą kolejną setkę poniżej 1700 dla tego zadania student dostaje 0.1p premii.

Ponadto, jeżeli pierwsze rozwiązanie studentów będzie zgłoszone na SKOSie przed opublikowaniem informacji o istnieniu rozwiązania wzorcowego, wartość zadania rośnie o co najmniej 4 punkty (decyzję podejmuje prowadzący, biorąc pod uwagę trudność rozwiązania). W przypadku powstania rozwiązania wzorcowego przed rozwiązaniem studenckim wartość również może wzrosnąć (zależnie od tego, jak trudne jest rozwiązanie wzorcowe). Mogą pojawić się wówczas wskazówki, odnoszące się do tego rozwiązania.