

Sztuczna inteligencja
Pracownia 3
Termin: 7 i 8 zajęcia

Prezent od Zajęczka: na pierwszej pracowni po Wielkanocy można:

1. Oddawać dwa ostatnie zadania P2 za pełne punkty
2. Uzyskać pełne punkty za P2.1 (ew. korzystając ze wskazówki, która jest poniżej)

Wskazówka do P2.1: Warto dać programowi dużo czasu przed restartem. W programie przechodzącym testy restart następował po $30M * N$ zmianach (30 razy wielkość planszy). Prawdopodobieństwo ruchów losowych było małe (około 1-2%). Pamiętaj, żeby wszystkie remisy rozstrzygać losowo, również losowo powinienes decydować, czy zajmować się wierszami, czy kolumnami.

Zadanie 1. (6p) W zadaniu tym powinienes dopisać moduł wnioskujący do obrazków logicznych (czyli procedurę, która analizując specyfikację np. wiersza jest w stanie dedukować informacje o zawartości poszczególnych pikseli, a następnie korzystając z tej wiedzy dedukować wartość innych pikseli rozważając specyfikacje kolumn, i tak dalej).

Testy do tego zadania są w większości powtórzeniem testów z zadania P2.1, taka sama jest też specyfikacja danych wejściowych i oczekiwanego wyniku. Inne limity czasowe. Możesz założyć, że w tym zadaniu samo wnioskowanie jest w stanie wydedukować wartość każdego piksela.

Zadanie 2. (6p) W tym zadaniu nadal rozwiązujesz obrazki logiczne, ale w rozwiązaniu powinien być użyty backtracking (testy do tego zadania będą dobrane w ten sposób, że najprawdopodobniej program z poprzedniego zadania nie będzie w stanie wypełnić wszystkich pikseli). Rozwiązanie modelowe, mieszczące się w limitach czasowych, przeplatało wnioskowanie i backtracking.

Zadanie 3. (1p) To zadanie jest łatwym przygotowaniem do zadania kolejnego. Rozważamy w nim problem Sudoku. Na stronie znajdziesz (prawie kompletny) program sudoku.py, który produkuje, dla konkretnej instancji łamigłówki sudoku, program w Prologu sudoku.pl, rozwiązujący tę łamigłówkę. Uzupełnij program sudoku.py (instrukcje znajdziesz w komentarzach w kodzie), tak aby przeszedł testy.

Opis formatu danych: opis łamigłówki składa się z 9 wierszy, każdy po 9 znaków. Znakami są cyfry (oznacza to, że w łamigłówce na tym miejscu jest wpisana cyfra), oraz kropki. Przykładowy opis łamigłówki (podobno trudnej):

```
3.....1
4..386...
.....1.4.
6.924..3.
..3.....
.....719
.....6
2.7...3..
```

Zadanie 4. (4p) W tym zadaniu powinienes napisać program, który rozwiązuje łamigłówkę burze (opis łamigłówki znajdziesz na początku wykładu W6). Twój program, dla każdej instancji zadania, powinien wypisywać program w SWI-Prologu, który rozwiązuje tę instancję (czyli wypisuje jedno rozwiązanie, jako listę kolejnych zer i jedynek).

Opis formatu danych: Pojedynczy przypadek testowy zawiera:

- Opis wierszy (czyli k liczb, w jednym wierszu)
- Opis kolumn (czyli m liczb, w jednym wierszu)
- Pewną liczbę deklaracji o wypełnionych polach, po jednej w wierszu. Deklaracje mają postać:

`<nr_wiersza> <nr_kolumny> <0_lub_1>`

Na stronie znajduje się również program storms.py¹, który (w trywialny sposób) rozwiązuje jeden przypadek testowy, co powinno wyjaśnić wątpliwości związane z formatem.

¹Choć przykład jest w Pythonie, rozwiązując zadanie możesz wybrać dowolny język.