

PROGRAMOWANIE W LOGICE

Parser SGML/XML

(Lista 8)

Przemysław Kobyłański

Wstęp

W pakiecie `library(url)` znajduje się predykat `parse_url(?URL, ?Atrybuty)`, który służy do analizy odnośnika URL.

Przykład użycia tego predykatu:

```
?- use_module(library(url)).  
true.
```

```
?- parse_url('http://cs.pwr.edu.pl/studenci.php', X).  
X = [protocol(http), host('cs.pwr.edu.pl'), path('/studenci.php')]
```

Jeśli odnośnik jest względny, to podając bazowy adres URL np. jako stałą `'http://localhost/'` można poprawnie rozpoznać predykatem `parse_url/3` adres lokalnego hosta:

```
?- parse_url('index.html', 'http://localhost/', X).  
X = [protocol(http), host(localhost), path('/index.html')].
```

O pakietach wykorzystywanych do rozwiązywania tej listy zadań możesz przeczytać na stronach:

- [http://www.swi-prolog.org/pldoc/doc_for?object=section\('packages/sgml.html'\)](http://www.swi-prolog.org/pldoc/doc_for?object=section('packages/sgml.html'))
- <http://www.swi-prolog.org/pldoc/man?section=url>

Zadania

Zadanie 1 (5 pkt)

Napisz predykat `serwery(+NazwaPliku, -ListaSerwerów)`, który analizuje plik HTML o nazwie `NazwaPliku` i dostarcza listę serwerów (hostów), do których prowadzą odnośniki w danym pliku.

Wskazówki

1. Użyj pakietów `library(sgml)`, `library(xpath)`, `library(url)`.
2. Użyj predykatu `findall/3` do znalezienia wszystkich serwerów (jeśli nie chcesz by serwery powtarzały się na liście, użyj odpowiednio predykatu `setof/3`).

Przykład

Założmy, że w pliku `index.html` znajduje się strona <http://cs.pwr.edu.pl/kobylanski/>.

Wówczas:

```
?- serwery('index.html', ListaSerwerów).
ListaSerwerów = ['ki.pwr.edu.pl', localhost, 'realmacsoftware.com',
                 'rekrutacja.pwr.edu.pl', 'studia.cs.pwr.edu.pl',
                 'wppt.pwr.edu.pl', 'www.pwr.edu.pl'].
```

Zadanie 2 (5 pkt)

Ściągnij archiwum [html.zip](#).

Po rozpakowaniu znajdziesz w katalogu `html` pliki `queens.pl`, `queens_in.html` i podkatalog z obrazkami: `empty.png` - dla pustego pola szachownicy, `queen.png` - dla pola, na którym stoi hetman.

W pliku `queens.pl` znajduje się predykat `queens(N, P)` rozwiązujący problem N hetmanów i dostarczający permutację P będącą jego rozwiązaniem.

W pliku tym znajduje się również fakt `size(N)` definiujący rozmiar problemu, dla którego tworzona będzie prezentacja.

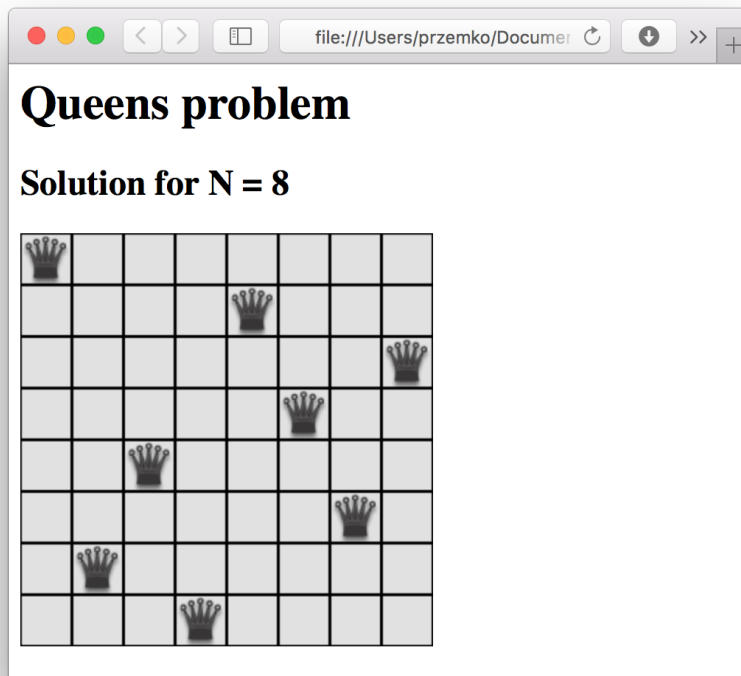
W pliku `queens_in.html` znajduje się szkielet dokumentu HTML wymagający uzupełnienia go odpowiednimi fragmentami kodu, tak by predykat `pwf_files/2` mógł na jego podstawie wygenerować poprawnie plik `queens.html`, prezentujący jedno rozwiązanie (nie zapomnij użyć predykatu `once/1` problemu hetmanów o zadanym w pliku `queens.pl` rozmiarze (fakt `size/1`)).

Polecenie

Dopisz w pliku `queens_in.html` brakujące fragmenty kodu tak aby po przetworzeniu go predykatem `pwf_files/2` prezentował rozwiązanie problemu hetmanów.

Wskazówki

1. Przeczytaj w dokumentacji pakietu `pwf` przykład objaśniający jak robić warunkowe generowanie fragmentów kodu HTML (przykład 5. `if-the-else effect`).



Rysunek 1: Prezentacja rozwiązania dla ośmiu hetmanów.

2. Dla uproszczenia kodu przyjmij, że permutacja $P = [P_1, P_2, \dots, P_N]$ koduje rozwiązanie w ten sposób, że P_i jest numerem kolumny, w której stoi hetman z i -tego wiersza szachownicy (licząc od góry).

Przykład

Na rysunku 1 przedstawiono widok przeglądarki wyświetlającej wygenerowany na podstawie pliku `queens_in.html` plik `queens.html` w przypadku gdy rozmiar planszy jest równy 8.