

بنام خدا

امین رشیدبیگی

شماره دانشجویی: 931009

درس : ساختمان داده

استاد : دکتر دهقان

Finite State Machine

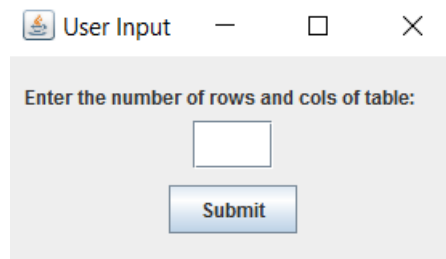
توضیحات کلی

این پروژه که اصطلاحاً FSM نام دارد، از کاربر ورودی را به صورت جدولی n در n دریافت می کند. این جدول عملاً همان ماتریس مجاورت برای گراف است. ویژگی هایی که این پروژه دارد، شامل نمایش لیست مجاورت گراف، تشخیص دنباله (sequence detector)، نمایش گراف به صورت شهودی و به صورت فایل png، تشخیص دور در گراف و همچنین حذف دور در گراف است.

کاربر هر تغییری که بخواهد اعمال کند، در یک Status Field که در قسمت سمت کاربر طراحی شده است می بیند.

راهنمای کاربر

با اجرا کردن برنامه، پنجره ای مطابق شکل زیر خواهید دید:



The image shows a window titled "User Input" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a text label "Enter the number of rows and cols of table:" followed by a single-line text input field. Below the input field is a blue button with the text "Submit" in white.

در این قسمت شما باید تعداد ستون ها و یا سطر های ماتریس مجاورت را وارد کنید.

بعد از زدن دکمه submit وارد پنجره زیر می شوید:

The screenshot shows a web-based state machine tool interface. At the top, there is a section titled "Fill the table" containing a 5x5 grid of input fields. Below this is a large empty rectangular area. An arrow labeled "1" points to this area. Underneath is a button labeled "Make The State Machine!". An arrow labeled "2" points to this button. Below the button are three input fields: "Enter the string to search:", "Start state:", and "Final States:". An arrow labeled "3" points to the "Enter the string to search:" field. Below these fields are four buttons: "Search", "Graph List", "Has Loop", and "Remove Loop". An arrow labeled "4" points to the "Remove Loop" button. At the bottom is another large empty rectangular area. An arrow labeled "5" points to this area.

شما ابتدا باید در قسمت 1، ورودی های خود را مشخص میکنید. توجه داشته باشید که اگر نخواهید از راسی به راس دیگر یالی موجود نباشد، مقدار اولیه جدول را به صورت - بگذارید. در قسمت 2 باید استیت آغازین و همچنین استیت های پایانی را مشخص کنید. توجه داشته باشید که اگر میخواهید چندین استیت پایانی داشته باشید، آن ها را با کاما جدا کنید. حال بر روی دکمه **make the state machine** کلیک کنید تا گراف مورد نظر ساخته شود. حال در قسمت 4 میتوانید رشته ای را وارد کنید و با زدن دکمه **search** در قسمت دکمه های 4، به وجود داشتن آن رشته در گراف پی ببرید. همچنین با زدن دکمه های دیگر، همانطور که از اسم آن های مشخص است، اعمالی انجام داده می شود و در تمامی آن ها شما در قسمت 5 برنامه خود، خروجی مورد نظر را دریافت خواهید کرد.

حذف و پیدا کردن دور

استفاده شده است. اما برای پیدا کردن دور، این DFS در این پروژه، برای تشخیص و همچنین حذف دور، به طور کلی، از الگوریتم را نظر grey، خاصیت unvisited و visited گراف علاوه بر خاصیت های node الگوریتم را گسترش داده ام. به صورتی که هر باشد، دور تشخیص داده می شود. همچنین برای حذف آن دور همانطور که مشخص است، grey دارد و اگر هر یالی، استیت مقصدش می رسد حذف می شود. grey آن یال که به

خروجی برنامه

شما با اجرا کردن این برنامه، یک خروجی png گراف خواهید دید. این ویژگی با استفاده از ابزار GraphViz به وجود آمده است. در زیر نمونه ای از این گراف را خواهید دید:

