

Autómatas de pila deterministas

Para esta tarea usted debe probar cierta cantidad de cadenas en los autómatas de pila deterministas dados, siguiendo el siguiente formato:

Entrada

La primera línea de la entrada tendrá un entero C , que indica el número de casos de prueba. La primera línea de cada caso de prueba contendrá cinco enteros E, T, F, S y C , donde E representa el número de estados en el autómata, T la cantidad de transiciones, F representa la cantidad de estados finales, S el estado inicial y C la cantidad de cadenas de prueba respectivamente. La siguiente línea contendrá F enteros, los cuales representan los estados finales del autómata. Luego vendrán T líneas, cada una con 2 enteros I y J y 3 cadenas; L, T y A , en donde I y J ($0 \leq I, J < E$) representan el estado origen y el estado destino de una transición respectivamente. L representa el carácter leído de la cinta de entrada en dicha transición, T representa el símbolo que se encuentra en el tope de la pila y A la acción que se debe realizar con el tope de la pila al final de dicha transición (el carácter utilizado para representar el fondo de la pila será siempre Z . Para representar el final de la cadena, la acción de desapilar o no tomar en cuenta el tope de la pila para la transición se utilizara el carácter \pounds `<alt+156>`). El alfabeto de la pila serán las letras mayúsculas. En el caso de la cadena A , los símbolos se apilan de derecha a izquierda (de la misma forma que en el programa JFlap, es decir, el nuevo tope de la pila será el carácter que se encuentre más a la izquierda). Luego vendrán C líneas, cada una con una cadena de entrada. Las cadenas de entrada podrán contener letras minúsculas y números (no necesariamente presentes en alguna transición).

La entrada debe ser leída desde el archivo “entradaP.in” el cual se encontrará en la carpeta del programa. Se considerará un bono si su programa provee la posibilidad de leer el archivo desde cualquier ruta mediante una interfaz gráfica (es decir, sin necesidad de escribir la ruta completa del archivo).

Salida

En la primera línea de cada caso de prueba debe mostrar la siguiente cadena “Caso G :", en donde G representa el número del caso de prueba (empezando en 1). Seguidamente C líneas en las cuales debe imprimir la palabra “Aceptar” en caso de que el autómata acepte dicha cadena o “Rechazar” en caso contrario.

Después de cada caso de prueba debe imprimir una línea en blanco.

La salida debe ser escrita en el archivo “salidaP.out”.

Ejemplo

Entrada	Salida
2	Caso 1:
3 5 1 0 5	Aceptar
2	Rechazar
0 0 1 Z XZ	Rechazar
0 0 1 X XX	Rechazar
0 1 0 X X	Aceptar
1 1 1 X £	
1 2 £ Z Z	Caso 2:
111101111	Aceptar
110111	Aceptar
011111	Rechazar
1010101	Rechazar
11011	Aceptar
4 6 1 0 5	
3	
1 2 b A £	
0 0 a Z AZ	
0 1 a A AAA	
1 0 a A AA	
2 3 £ Z Z	
2 2 b A £	
aabbb	
aaaabbbbbbb	
c1bbb	
abbb	
aaaaaabbbbbbbbbb	

Condiciones de entrega:

- La tarea puede ser entregada de forma individual o en grupos de dos personas.
- Lenguajes a utilizar: C++, C# o JAVA.
- Los programas generados deben seguir al pie de la letra el formato de las entradas y salidas indicadas en el enunciado (incluyendo espacios). No se corregirán proyectos que no lean correctamente la entrada o den como resultado una salida con un formato incorrecto.

- La fecha límite para entregar el proyecto es el jueves 21/06/2011. A más tardar ese día, deben enviar por correo todos los códigos fuente de su programa y un ejecutable en un archivo .rar a ambos preparadores.
- La fecha y horario para la defensa de la tarea será publicada posteriormente. La defensa de la tarea es obligatoria. En caso de no asistir a la defensa, su tarea será ponderada con la calificación mínima.
- Las copias serán severamente penalizadas según lo establecido en la Ley de Universidades. Se anima a la discusión pero se prohíbe la copia de proyectos. Cualquier proyecto entregado debe ser fruto de su propio trabajo.

Francisco Suárez / Junio 2011