

UTN FRBA – SSL – Examen Final – 2019-07-29

| | | | | | |
|-------------------|--|---------|--|-------|--|
| Apellido, Nombre: | | Legajo: | | Nota: | |
|-------------------|--|---------|--|-------|--|



- Resuelva el examen en tinta y en esta hoja; no se aceptan hojas adicionales.
- Para los ítems de *una mejor respuesta*, marcados con una círculo (○), tilde (✓) sólo una opción, la mejor.
- Para los ítems de *respuestas múltiple*, marcados con un caja (□), tilde (✓) todas las respuestas correctas.
- Durante el examen no se responde consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. (1 punto) Indique y justifique el valor de verdad de la siguiente afirmación:
Las operaciones *unión*, *intersección*, y *complemento* son parte de las *regex*.
2. (2 puntos) Escriba un ejemplo de una expresión C con por lo menos un error semántico. Justifique.
3. (1 punto) Indique cuál es la entrada a la parte (o proceso) de síntesis de un compilador.
4. (1 punto) Indique la secuencia de algoritmos para obtener un AFD a partir de una ER.
5. (4 puntos) En la siguiente tabla tilde para cada lenguaje el o los autómatas capaces de reconocerlo:

| Lenguaje | AFD | AFN | AP | MT |
|---|-----|-----|----|----|
| Constantes enteras de C | | | | |
| $\{a^n b^n c^n \mid n \geq 1\}$ | | | | |
| Sentencias de iteración de C | | | | |
| $\{a^n b^t \mid n \geq 1 \wedge t \geq 1\}$ | | | | |

6. Sea la expresión ++automata->finales[i]
 - a. (1 punto) Reescriba la expresión con paréntesis redundantes que expliciten la precedencia y la asociatividad.
 - b. (Punto Extra) Escriba las declaraciones para que sea semánticamente correcta.

1. Una Resolución

1. Falso. Contiene la operación unión (\cup), contiene solo un caso muy particular del complemento ($\bar{}$), y no contiene la intersección.

2. ++42.

El operador ++ requiere un *l-value modifiable* y las constantes no lo son.

3. La *representación intermedia*.

4. a. Algoritmo de Thompson,

b. Algoritmo de Clausuras- ϵ o de Construcción de Subconjuntos,

c. y, opcionalmente, Algoritmo de Minimización.

5.

| AFD | AFN | AP | MT |
|-----|-----|----|----|
| X | X | X | X |
| | | | X |
| | | X | X |
| X | X | X | X |

1. a. (++((automata->finales)[i]))

b.

```
typedef int Estado;
typedef struct {Estado *finales;} Automata;
Automata *automata;
int i;
```

v1.0.0, 2019-07-31