*Departamento de Tecnologías de la Información*

*Área de Ciencias de la Computación e Inteligencia Artificial*

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^y \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^y \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas $<M,w>$ tales que $M$ es la codificación de una máquina de Turing y $w$ es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^y \le x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

*Departamento de Tecnologías de la Información*

*Área de Ciencias de la Computación e Inteligencia Artificial*

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento (*x*+1) el caso base de la recursión es *x*=0.

$$Log(x+1, n) = y \mid n^y \le x + 1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

*Departamento de Tecnologías de la Información*

*Área de Ciencias de la Computación e Inteligencia Artificial*

Universidad de Huelva

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \ | \ n^y \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

*Departamento de Tecnologías de la Información*

*Área de Ciencias de la Computación e Inteligencia Artificial*

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^{y} \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas *<M,w>* tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^y \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

Universidad de Huelva

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento ($x$+1) el caso base de la recursión es $x$=0.

$$Log(x+1, n) = y \mid n^{y} \le x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?

*Departamento de Tecnologías de la Información*

*Área de Ciencias de la Computación e Inteligencia Artificial*

# Modelos Avanzados de Computación

# Examen de septiembre

**EJERCICIO 1 (1 punto)**

Enuncie y demestre el Lema de Bombeo para Autómatas Finitos.

**EJERCICIO 2 (1.5 puntos)**

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde L es el símbolo inicial.

| | | | |
|---|---|---|---|
| L → NM F | MT → SM T | T → **producto** | RP → **rpar** |
| L → LP LC | MT → SM L | F → ML T | NM → **num** |
| L → T MT | T → NM F | LC → L RP | SM → **plus** |
| L → **producto** | T → LP LC | LP → **lpar** | ML → **mul** |

Verifique que la cadena "**num mul lpar producto plus num mul producto rpar**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

**EJERCICIO 3 (2 puntos)**

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos {A,B,C}. Por ejemplo, para la entrada "#AABCA~~bbbb~~..." devuelve en la cinta "#AABCAAABCA~~bb~~..".

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

**EJERCICIO 4 (1.5 puntos)**

Sea $HALT_{TM}$ el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando). Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

**EJERCICIO 5 (2 puntos)**

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Log*(*x+1,n*), que calcula el logaritmo en base *n* de un número entero, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento (*x*+1) el caso base de la recursión es *x*=0.

$$Log(x+1, n) = y \mid n^y \leq x+1 < n^{y+1}$$

**EJERCICIO 6 (1 punto)**

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

**EJERCICIO 7 (1 punto)**

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?