Lenguajes de Programación

Unidad I

“Conceptos de Programación”

Rogelio Ferreira Escutia

Contenido

1) Introducción a la Programación
2) Lenguajes de Alto y Bajo Nivel
3) Generaciones de Lenguajes de Programación
4) Paradigmas de Programación
5) Algoritmos
6) Expresión de un Algoritmo
7) Técnicas de Descripción Formal

1) Introducción a la Programación

Conceptos

- **"Un programa computacional es un conjunto de instrucciones detalladas, paso a paso, que le indican a la computadora cómo resolver un problema o realizar una tarea"[38].**
- **"Un lenguaje de Programación es un sistema notacional para. describir computaciones en una forma legible tanto para la máquina como para el ser humano"[45].**

- **Las instrucciones que integran un programa computacional se denominan código, debido a que anteriormente las instrucciones de programa para las computadoras de primera generación se ingresaban como códigos binarios.**

- **En la actualidad, el código de un programa contiene palabras familiares en inglés**

38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición, 45) Lenguajes de Programación, Kenneth C. Louden.

Partes de un Programa

- Cada instrucción de un programa computacional está integrada por palabras clave y parámetros que se unen mediante reglas de sintaxis.
- Una palabra clave o comando es una palabra que tiene un significado predefinido para el compilador o intérprete, que traduce cada línea del programa a lenguaje máquina.
- Las palabras clave se combinan con parámetros específicos, que proporcionan instrucciones más detalladas para que las realice la computadora.
- Las palabras clave y los parámetros se combinan con signos de puntuación de acuerdo con una serie de reglas llamadas sintaxis.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

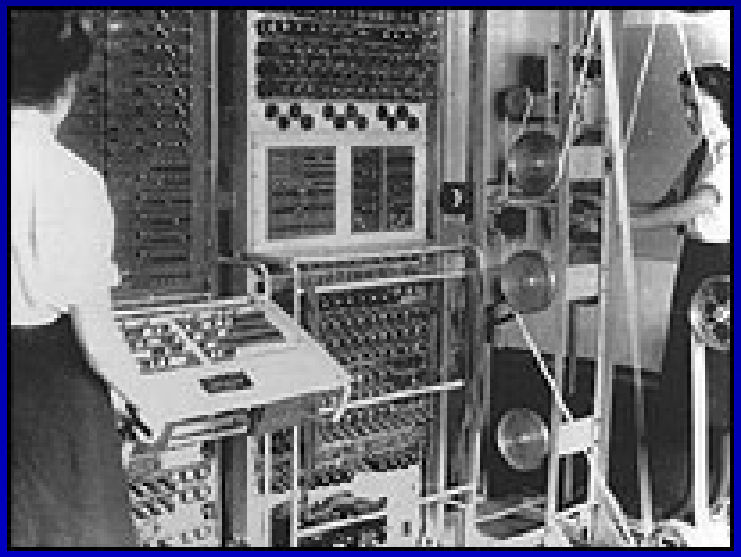
Primer Programador

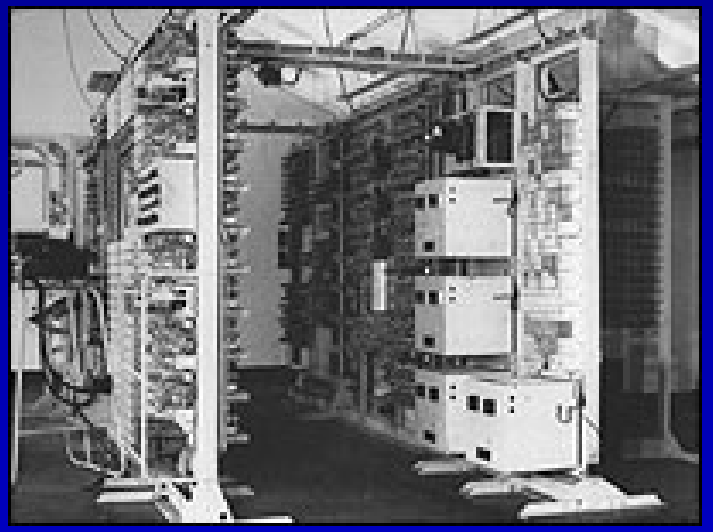
- Se considera a Augusta Ada Byron (1815-1852) como el primer programador de la historia.
- Hija del poeta Lord Byron y de la matemática Annabella Milbanke Byron trabajó junto con Charles Babbage en la máquina analítica, de los cuales realizó apuntes sobre como programar dicha máquina, los cuales se conservaron y comprobaron que ella realizó los primeros programas.

- The TechWeb, http://www.techweb.com/encyclopedia/defineterm?term=ADA&x=0&y=0 - The Ada Picture Gallery, http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/pictures.html

Primera Computadora

- Colossus es la primer computadora que trabaja a nivel de 2 bits.
- Fue construida en Inglaterra, durante la Segunda Guerra Mundial y se encargaba de leer información que se interceptaba a los alemanes.
- La información se introducía por medio de tarjetas perforadas, y se que lograba leer 5000 caracteres por segundo.



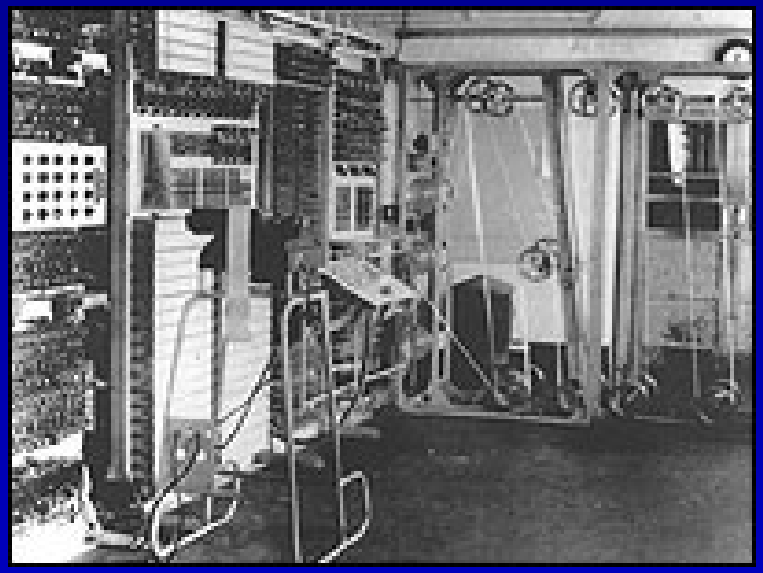


- BBC News Technology, http://news.bbc.co.uk/1/hi/technology/3754887.stm

Primera Computadora

- **Colossus lograba procesar 100 operaciones boleanas por cada una de las 5 cintas de entrada de datos.**

- **Estas operaciones eran programadas por medio de una circuiteria alambrada como la computadora ENIAC, no existía los programas almacenados como tal.**



- BBC News Technology, http://news.bbc.co.uk/1/hi/technology/3754887.stm

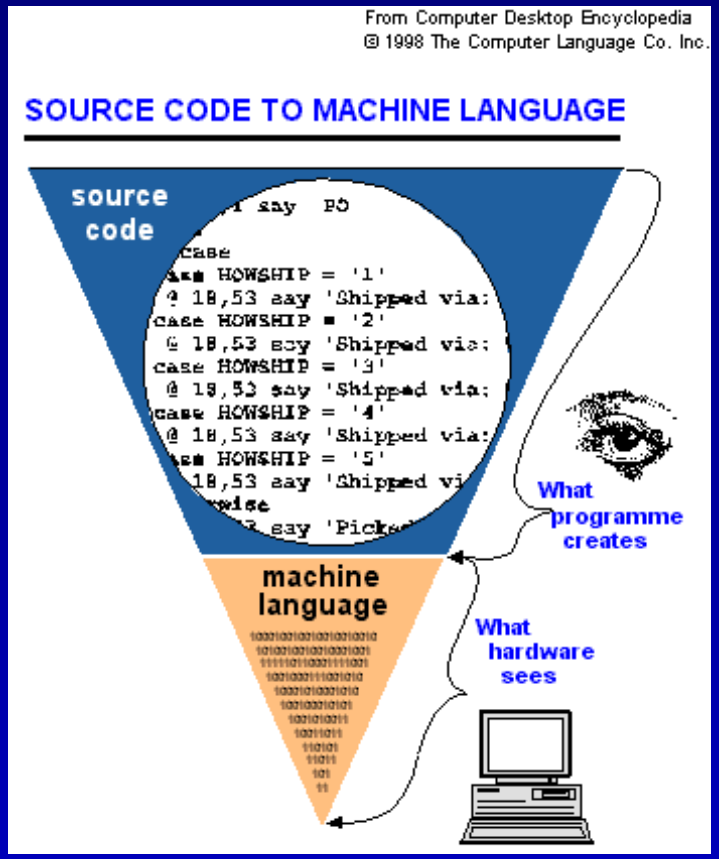
2) Lenguajes de Alto y Bajo Nivel

Clasificación de Lenguajes

- Los lenguajes se clasifican de diferentes maneras. Pueden dividirse en 2 categorías principales:

1) Lenguaje de Bajo Nivel

- Incluyen comandos que son específicos para una familia de microprocesadores determinados. Se requiere que un programador escriba instrucciones en lenguaje de bajo nivel para ese hardware específicamente, estas dependen del tipo de procesador, registros y direcciones de memoria.



- 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición, The TechWeb, http://news.bbc.co.uk/1/hi/technology/3754887.stm

Clasificación de Lenguajes

2) Lenguaje de Alto Nivel

- Proporciona un nivel de abstracción que oculta el lenguaje ensamblador o de bajo nivel y proporciona palabras y gramática de comandos que son mas parecidos al lenguaje humano

- Lenguajes de alto nivel como Cobol, Basic, Java y C, facilitan el proceso de programación al reemplazar cadenas ininteligibles de 1´s y 0´s, o comandos de ensamblador, con comandos mas sencillos como Print y Write.

- Los comandos de lenguaje de alto nivel eliminan muchas líneas de código al incluir en un solo comando de alto nivel varios comandos de bajo nivel

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

3) Generaciones de Lenguajes de Programación

Generaciones de Lenguajes

- **Las primeras computadoras fueron programadas sin lenguajes de programación, los técnicos sólo reconectaban los cables de los circuitos de la computadora para prepararla para varias tareas de procesamiento.**

- **La idea de almacenar programas en la memoria de la computadora abrió el camino a los lenguajes de programación que permitieron a los programadores escribir una serie de comando y cargarlos en la computadora para su ejecución.**

- **Los primeros lenguajes de programación eran primitivos, pero en el transcurso de varias décadas, evolucionaron hasta dar paso a los lenguajes de hoy en día.**

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Primera Generación

- El lenguaje máquina fue el primer lenguaje disponible para programar computadoras.
- Un lenguaje máquina ofrece un conjunto de comandos representados como series de 1´s y 0´s, que corresponden al conjunto de instrucciones que se encuentra grabado en los circuitos de un microprocesador.
- Un lenguaje máquina es específico de una familia de CPU o microprocesadores.
- Aunque el lenguaje máquina aún funciona en las computadoras actuales, los programadores rara vez los usan para escribir programas.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Segunda Generación

- El lenguaje ensamblador permite a los programadores emplear palabras de comando abreviadas, como LDA (“Load”=cargar) en lugar de los 1´s y 0´s usados en el lenguaje máquina.
- Un lenguaje ensamblador se clasifica como lenguaje de bajo nivel, porque es específico de una máquina (cada comando de lenguaje ensamblador tiene una correspondencia uno a uno con una instrucción de lenguaje máquina.
- Un lenguaje ensamblador es útil cuando un programador quiere manipular directamente lo que sucede en el nivel del hardware.
- En la actualidad, los programadores usan lenguaje ensamblador para escribir software de sistema, como compiladores, sistemas operativos y controladores de dispositivos.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Tercera Generación

- **Los lenguajes de esta generación emplean palabras de comandos fáciles de recordar, como PRINT e INPUT, que toman el lugar de varias líneas de código de lenguaje ensamblador, o cadenas interminables de 0´s y 1´s de lenguaje máquina.**

- **Se creía que con estos lenguajes se eliminarían los errores de programación, y aunque los errores fueron menos frecuentes y el tiempo de desarrollo disminuyó significativamente, pero los programadores también cometían errores, por lo que era necesario seguir desarrollando nuevos lenguajes de programación.**

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Cuarta Generación

- Su característica importante es que tienen mayor parecido con los lenguajes humanos o lenguajes naturales.
- De esta generación surge SQL y RPG-1, los cuales eliminan gran parte de la puntuación y las reglas gramaticales estrictas, que hacía complicados a los lenguajes de tercera generación.
- En la actualidad, los lenguajes de cuarta generación se usan en aplicaciones de base de datos. Un solo comando de SQL, reemplaza muchas líneas de código de tercera generación.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Quinta Generación

- En 1982, un grupo de investigadores japoneses empezaron a trabajar en proyecto que denominaron de quinta generación que usaba Prolog, que es un lenguaje declarativo.
- Por lo tanto, algunos expertos clasificaron a Prolog y otros lenguajes declarativos como de quinta generación.
- Otros expertos están en desacuerdo y piensan que los lenguajes de quinta generación son los que permiten a los programadores usar herramientas gráficas o visuales para construir programas en lugar de escribir líneas de código.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

4) Paradigmas de Programación

Paradigmas

- Además de clasificarse por nivel y generación, los lenguajes de programación se clasifican por paradigma.
- Los programadores afrontan los problemas de diferentes maneras, por lo cual seleccionan el paradigma de programación mas adecuado.
- La frase “Paradigma de Programación” alude a la manera de cenceptualizar y estructurar las tareas que realiza una computadora.
- Existen numerosos paradigmas de programación y no son mutuamente excluyentes.
- Los lenguajes de programación se ordenan mediante los paradigmas de programación que soportan.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Clasificación según los Paradigmas

Paradigma	Lenguajes	Descripción
Procedural	BASIC, Pascal, COBOL, FORTAN, Ada	Destaca los algoritmos lineales, paso a paso, que proporcionan a la computadora las instrucciones para resolver un problema o realizar una tarea.
Orientado a Objetos	Smalltalk, C++, Java	Formula programas como una serie de objetos y métodos que interactúan para realizar una tarea específica.
Declarativo	Prolog	Se concentra en el uso de hechos y reglas para describir un problema.
Funcional	LISP, Scheme, Haskell	Destaca la evaluación de expresiones, llamadas funciones.
Orientado a Eventos	Visual Basic, C#	Se concentra en la selección de elementos de interfaz de usuario y la definición de rutinas de manejo de eventos que se disparan con diversas actividades del ratón o del teclado.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

5) Algoritmos

Algoritmos

- **Un algoritmo es un conjunto de pasos para realizar una tarea, que puede escribirse y aplicarse.**

- **Una característica importante de un algoritmo correctamente formulado es que si se siguen cuidadosamente los pasos, se tendrá la garantía de que se completará la tarea para la que se diseñó.**

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

6) Expresión de un Algoritmo

Expresión de un Algoritmo

- **Existen varias maneras, como son lenguaje estructurado, seudocódigo, diagramas de flujo, etc.**

- **Estas herramientas no son lenguajes de programación y por lo tanto una computadora no tiene capacidad para procesarlos, su propósito es proporcionar una manera de documentar sus ideas para el diseño de programas.**

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Lenguaje Estructurado

- **Es un subconjunto de un lenguaje, típicamente el inglés, con una selección limitada de estructuras de frases que reflejan las actividades de procesamiento.**

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

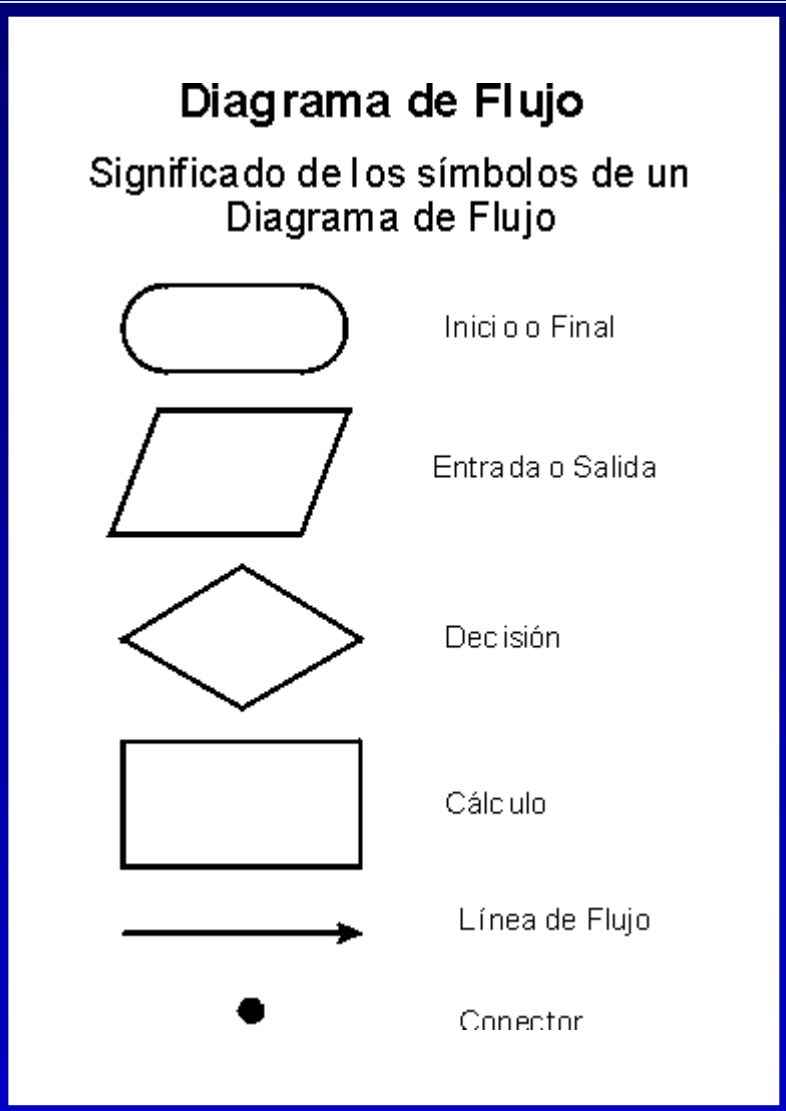
Seudocódigo

- Es un sistema de notación para algoritmos que se han descrito como una mezcla de lenguaje común y su lenguaje de programación favorito.
- El seudocódigo esta menos formalizado que el lenguaje estructurado de modo que la estructura y la redacción dependen del usuario.
- Cuando se escribe seudocódigo, se permite incorporar palabras de comandos y sintaxis del lenguaje computacional que se pretende usar para el programa actual.

• 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

Diagrama de Flujo

- **Es una representación gráfica de la manera en que una computadora debe de pasar de una instrucción a la siguiente cuando realiza una tarea.**



- 38) Nuevas Perspectivas: Conceptos de Computación, June Jamrich Parsons y Dan Oja, Thomson Editores 6° Edición

7) Técnicas de Descripción Formal

Técnicas de Descripción Formal

Redes de Petri (RP)

- **Representación gráfica natural de concurrencia.**
- **Definición formal.**
- **Modelo ejecutable asociado.**
- **Compatible con Análisis Automatizado.**
- **Es un grafo bipartita.**
- **Contiene plazas de entrada y arcos de salida**

Técnicas de Descripción Formal

Componentes de las Redes de Petri

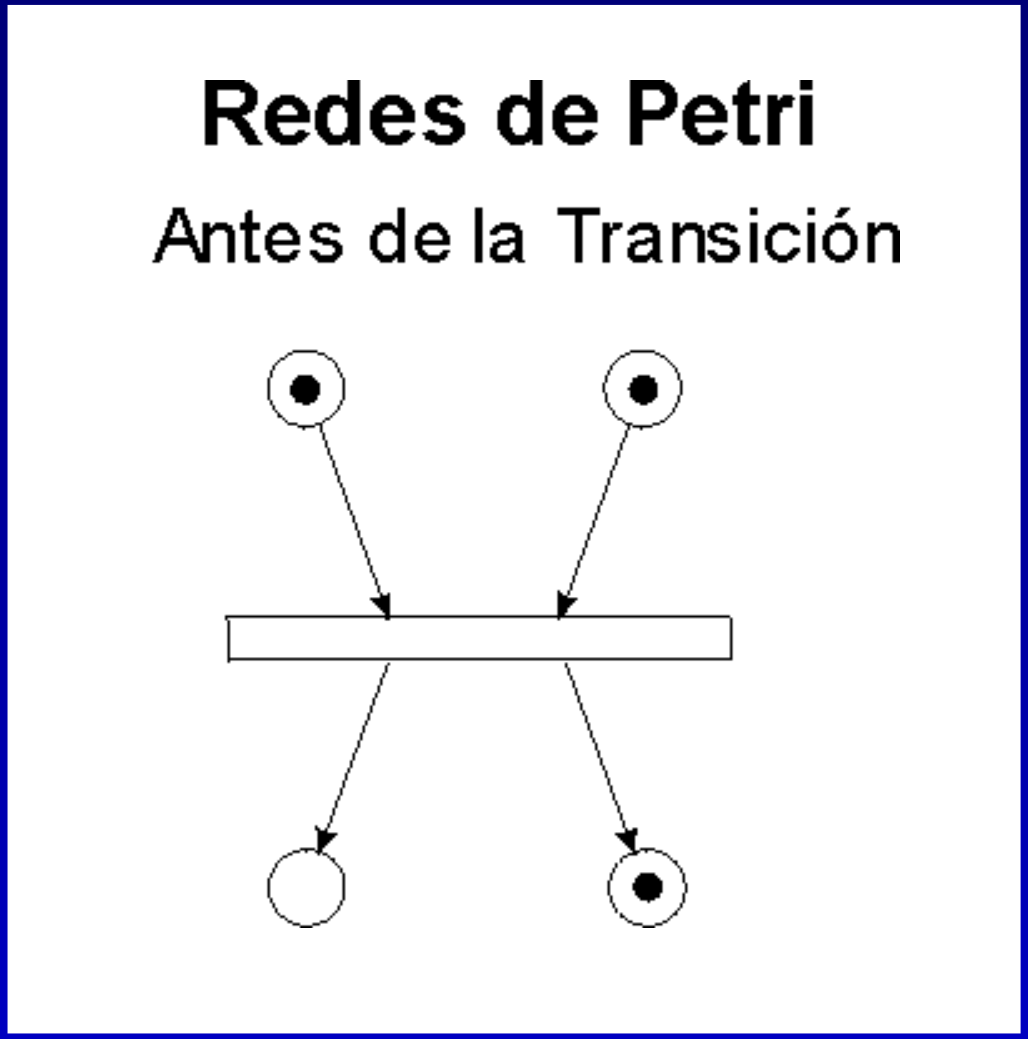
- **Plazas representadas con círculos que pueden contener Tokens.**
- **Transiciones representadas por rectángulos.**
- **Arcos dirigidos, indicando el efecto del disparo de transiciones que afecta plazas vecinas.**

Técnicas de Descripción Formal

Funcionamiento de la Redes de Petri

- **La transición dispara si todas las plazas de entrada contienen al menos un token.**
- **Al disparar se quita un token de cada plaza de entrada y se agrega un token a cada plaza de salida.**
- **Una Red de Petri se ejecuta al establecer un marcado inicial y en tiempos sucesivos una o mas transiciones habilitadas se seleccionan para ser disparadas.**

Técnicas de Descripción Formal

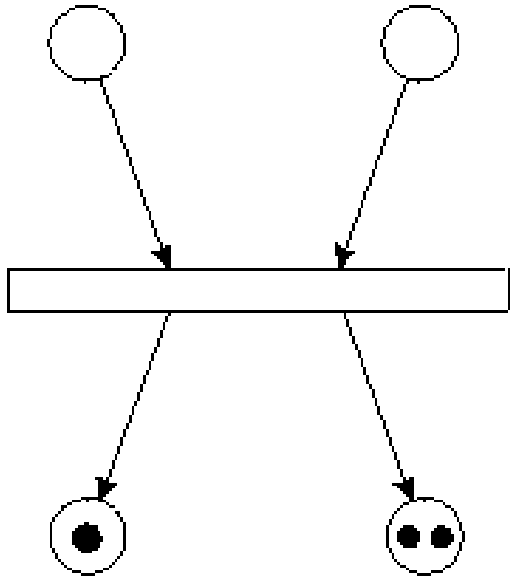


- Apuntes de la materia “Lógica Computacional” – Dr. Juan Frausto Solís – ITESM Campus Morelos 1997

Técnicas de Descripción Formal

Redes de Petri

Después de la Transición



- Apuntes de la materia “Lógica Computacional” – Dr. Juan Frausto Solís – ITESM Campus Morelos 1997

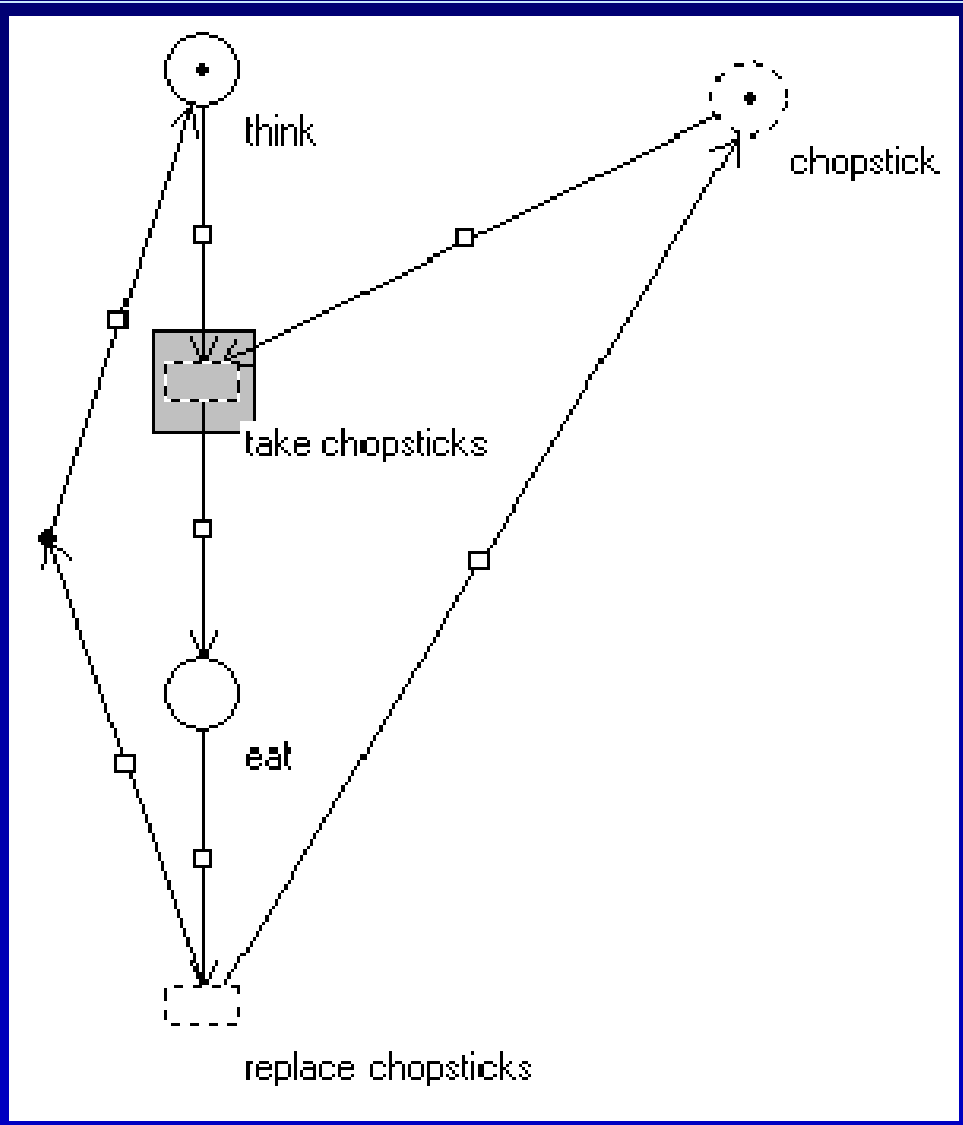
Técnicas de Descripción Formal

Problema de los Filósofos (ejemplo con 1)

Hay 1 filósofo comiendo en la mesa.

Para comer es necesario que esté pensando y estén disponibles los cubiertos(o palillos chinos).

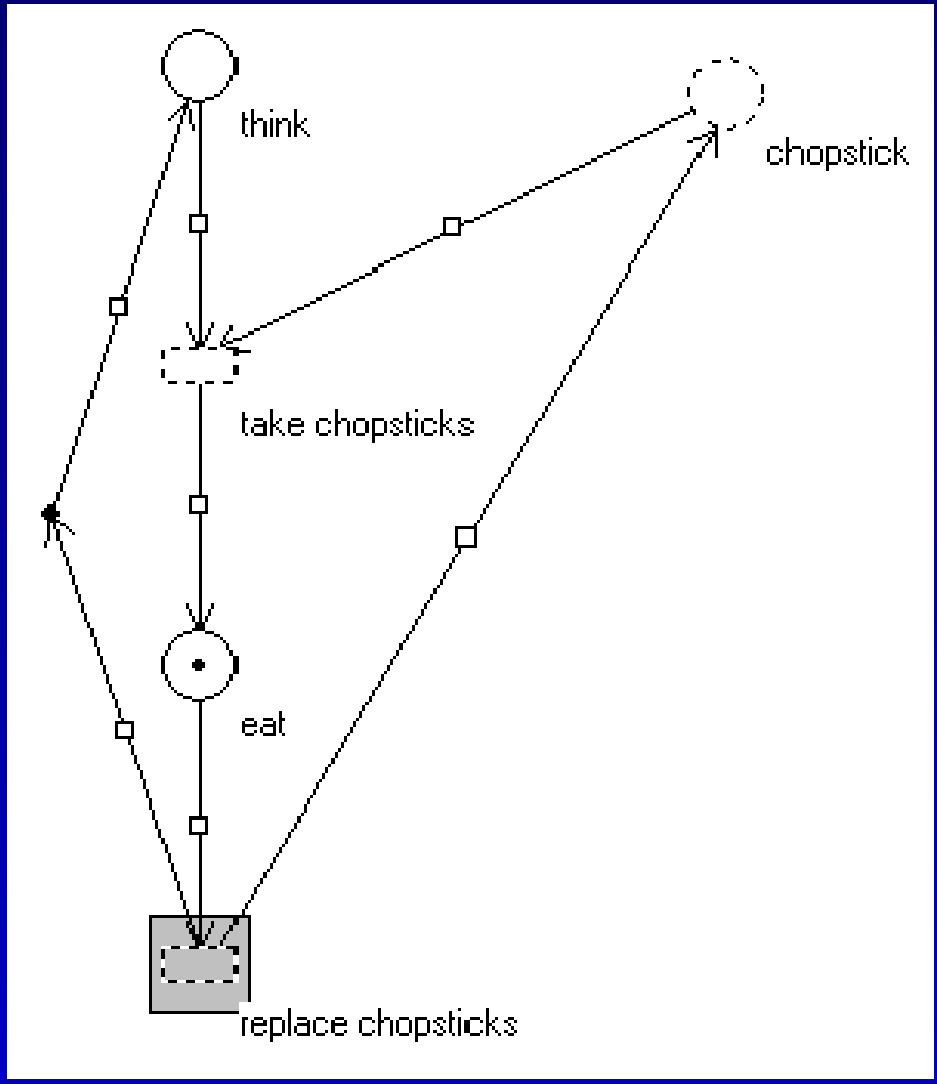
Aquí el filósofo esta pensando



- Simulador de Redes de Petri DNAnet (ejemplo que viene en el Software)

Técnicas de Descripción Formal

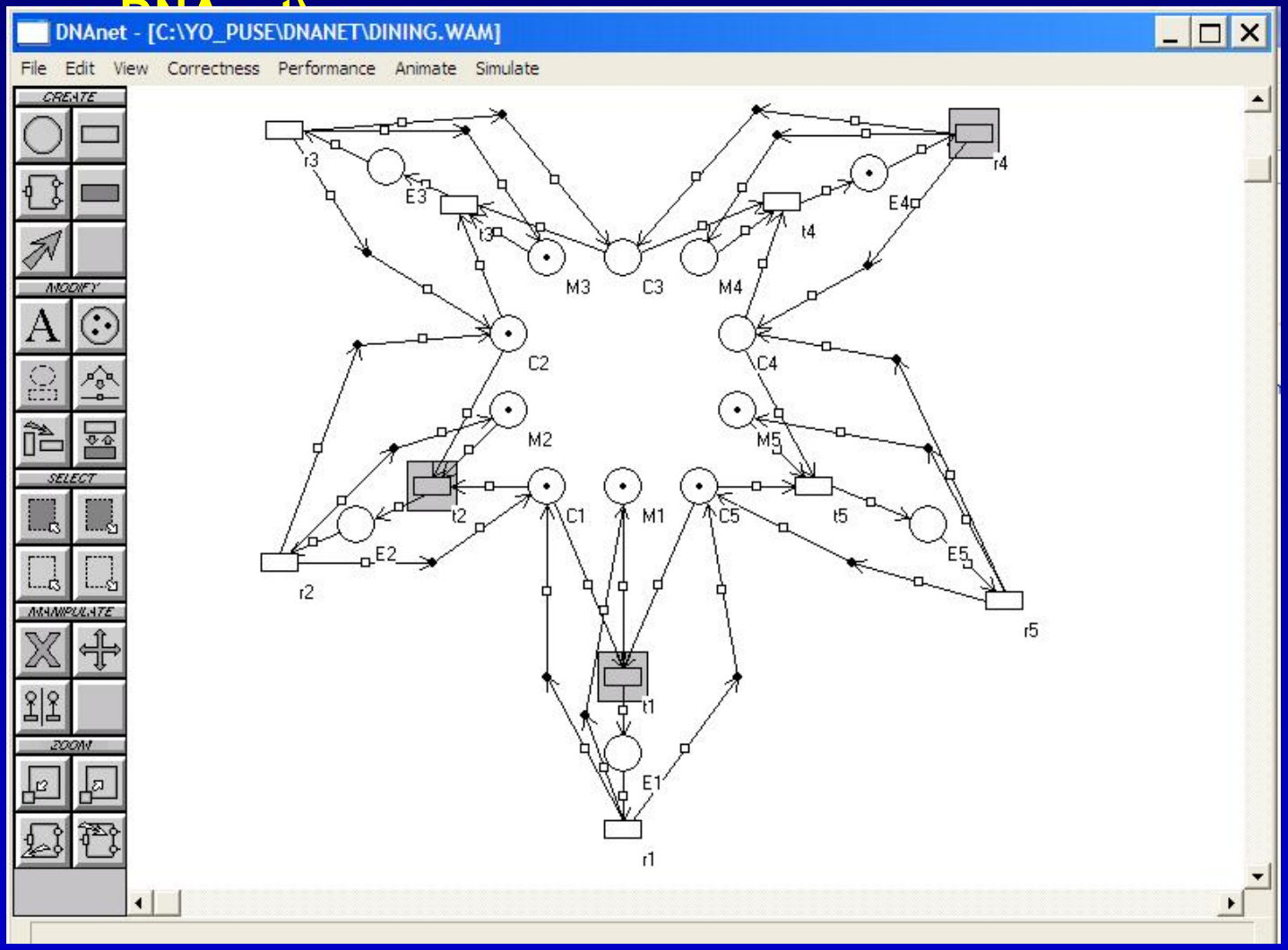
Aquí el filósofo esta comiendo.



- Simulador de Redes de Petri DNAnet (ejemplo que viene en el Software)

Simulador de Redes de Petri

Problema de los 5 Filósofos (en el Simulador DNAnet)



- Simulador de Redes de Petri DNAnet (ejemplo que viene en el Software)

Técnicas de Descripción Formal

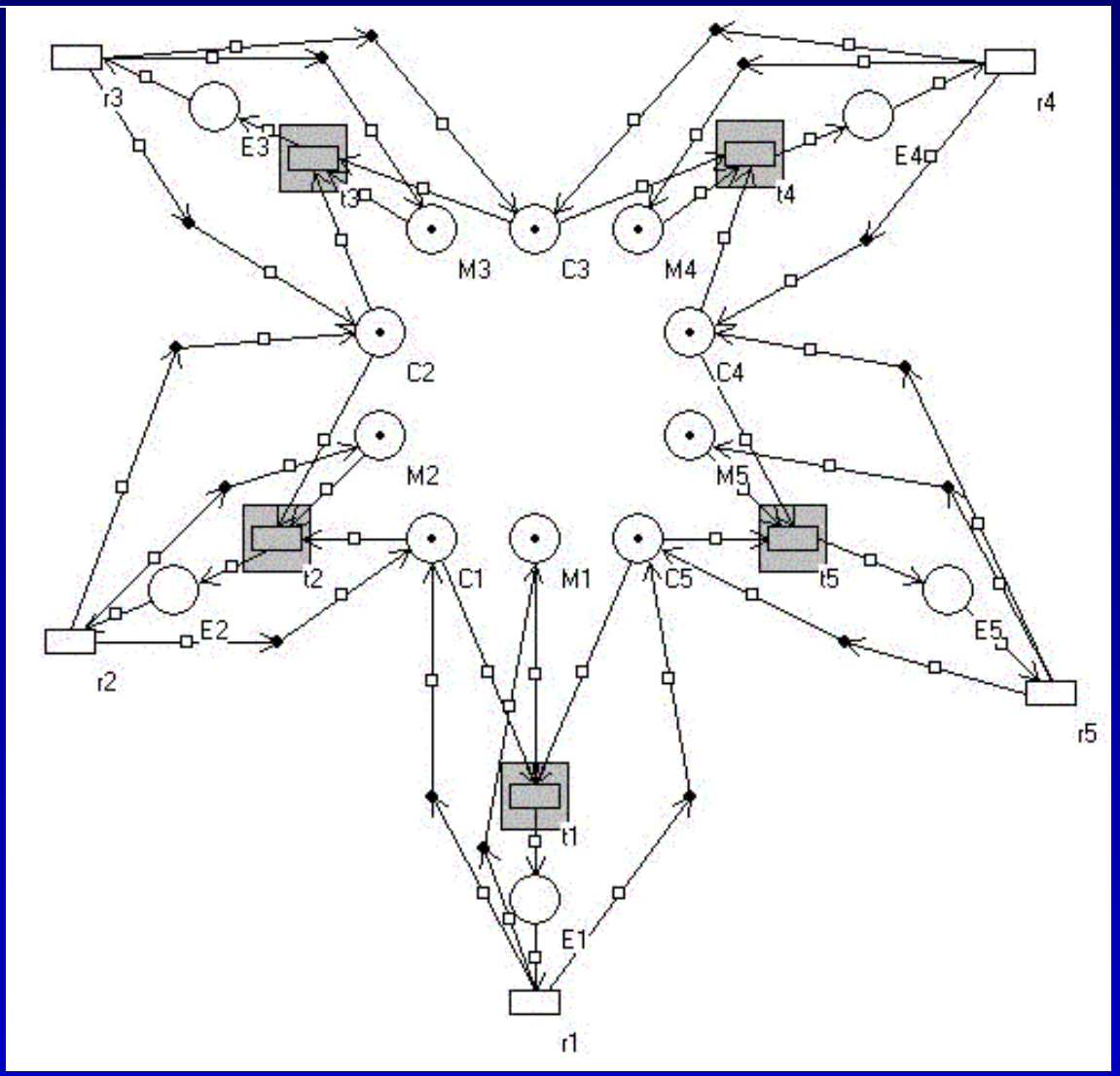
Problema de los Filósofos (ejemplo con 5)

Hay 5 filósofos comiendo en la misma mesa.

Para comer es necesario que cada filósofo agarre 2 tenedores.

Cuando un filósofo come, no pueden comer los que se encuentran a su lado.

- **M1 .. M5 Filósofo en espera de comer**
- **C1 .. C5 Tenedores**
- **E1 .. E5 Filósofo comiendo**
- **t1..t5 – r1..r5 transiciones**

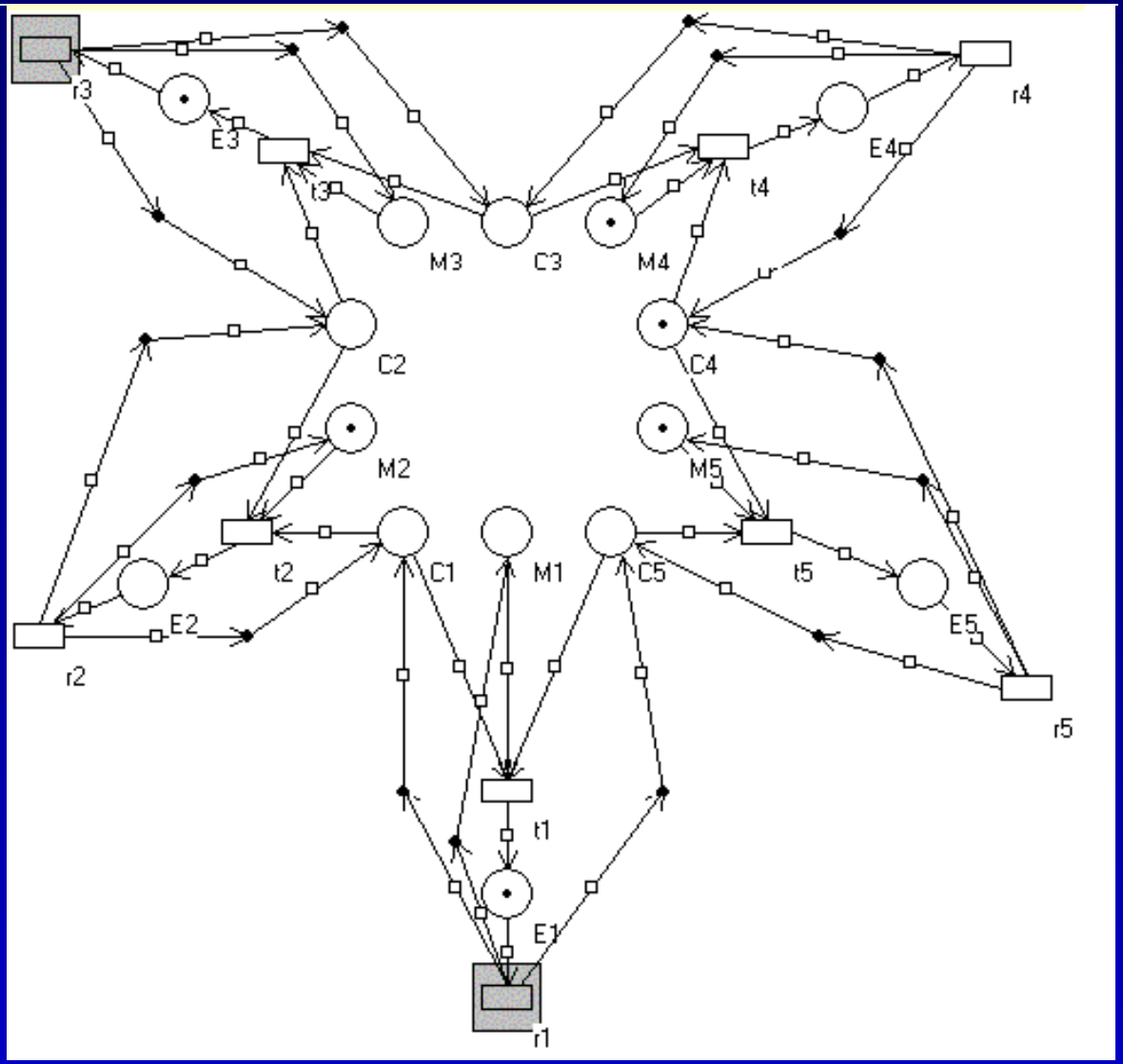


- Simulador de Redes de Petri DNAnet (ejemplo que viene en el Software)

Técnicas de Descripción Formal

Filósofo 1 y 3 comiendo (E1 y E3)

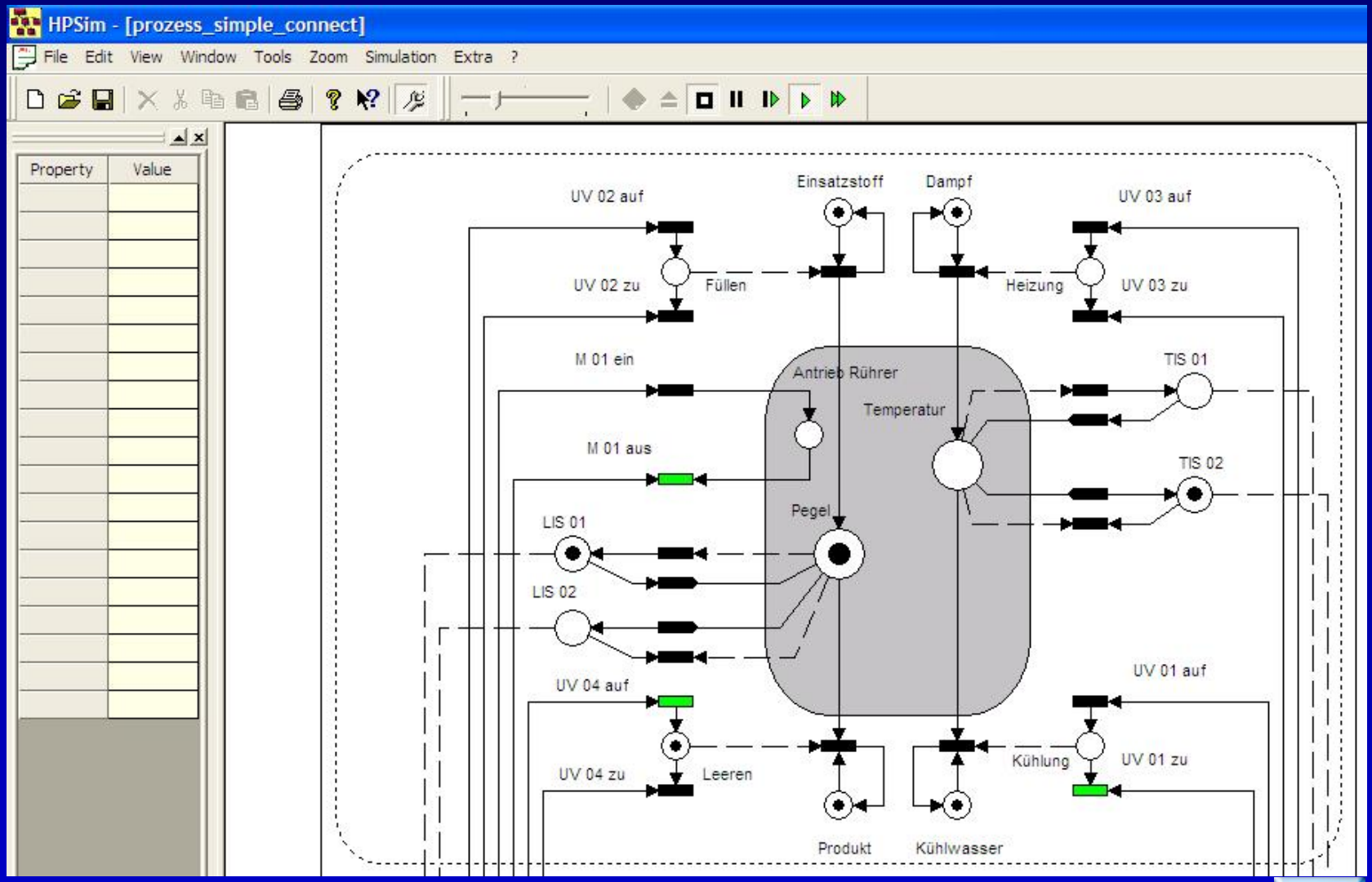
Filósofos 2, 4 y 5 no pueden comer.



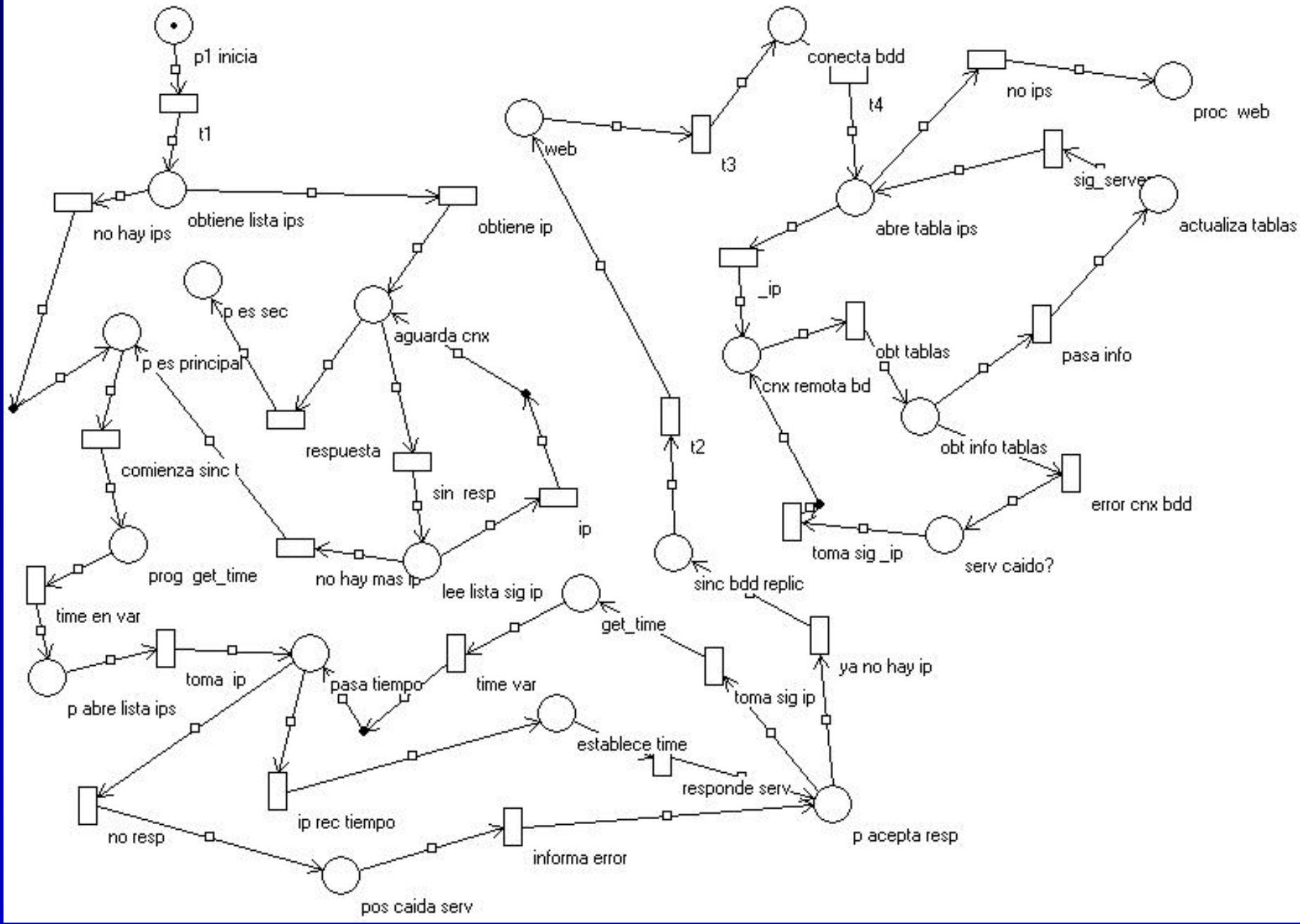
- Simulador de Redes de Petri DNAnet (ejemplo que viene en el Software)

Simulador de Redes de Petri

Simulador HPSim



- Simulador de Redes de Petri HPSim (ejemplo que viene en el Software)


p1 inicia
t1
obtiene lista ips
no hay ips
obtiene ip
p es sec
aguarda cnx
p es principal
respuesta
comienza sinc t
sin resp
ip
prog get_time
no hay mas ip
time en var
p abre lista ips
toma ip
pasa tiempo
time var
lee lista sig ip
get_time
toma sig ip
establece time
responde serv
ip rec tiempo
no resp
informa error
pos caida serv
p acepta resp
ya no hay ip
web
t3
conecta bdd
t4
no ips
proc web
sig_serve
abre tabla ips
actualiza tablas
_ip
obt tablas
pasa info
cnx remota bd
obt info tablas
error cnx bdd
toma sig _ip
serv caido?
t2
sinc bdd replic

Unidad I – Conceptos de Programación

Rogelio Ferreira Escutia

Instituto Tecnológico de Morelia
Departamento de Sistemas y Computación

Correo: ***rogelio@itmorelia.edu.mx***
rogeplus@gmail.com

Página Web: ***http://sagitario.itmorelia.edu.mx/~rogelio/***
http://www.xumarhu.net/

Twitter: ***http://twitter.com/rogeplus***
Facebook: ***http://www.facebook.com/groups/xumarhu.net/***