>>> IF013 - Fundamentos Teóricos de Informática
>>> Licenciatura de Sistemas - UNPSJB - Sede Trelew

Name: Celia Cintas[†], Pablo Navarro[‡], Samuel Almonacid[§]
Date: September 11, 2017

[†] cintas@cenpat-conicet.gob.ar, cintas.celia@gmail.com, @RTFMCelia
[‡] pnavarro@cenpat-conicet.gob.ar, pablo1n7@gmail.com
[§] almonacid@cenpat-conicet.gob.ar, almonacid.samuel.tw@gmail.com

>>> Unidad 3

1. Gramáticas sensibles al contexto. Autómatas acotados linealmente.
2. Propiedades formales de los lenguajes sensibles al contexto.
3. Maquinas de Turing. Reconocedoras y Generadoras.
4. Tesis Turing-Church. Limitaciones de la Computabilidad Efectiva. Maquinas de Turing No Determinísticas.
5. Problema de detención de Máquina de Turing.
6. Reductibilidad.
7. Redes de Petri.
8. Lenguajes y gramáticas sin restricciones.
9. Enumerabilidad de los lenguajes sensibles al contexto y argumento diagonal de Turing.

>>> Máquina de Turing y Computabilidad

Definición

Sea Σ un alfabeto y sean Y, N dos símbolos $\notin \Sigma$. Un lenguaje $L \subseteq \Sigma^*$ se dice decidible por una máquina de Turing (o Turing-decidible) si y sólo si la función $X_L : \Sigma^* \rightarrow \{Y, N\}$ es Turing-computable, y para cada $w \in \Sigma^*$, dá una respuesta acorde con:

$$X_L(w) =_{def} \begin{cases} Y si\ w \in L \\ N si\ w \notin L \end{cases}$$

Definición

Una máquina de Turing no determinista es una quintupla $T = (S, \Sigma, \delta, s_0, F)$ donde:

S es el conjunto de estados, $S \neq \emptyset$

Σ es el alfabeto de entrada.

$\delta : S \times \Sigma \rightarrow P(S \times \Sigma \times \{I, D, N\})$.

s_0 es el estado inicial, $s_0 \in S$.

F es el conjunto de estados finales, $F \subseteq S$.

>>> Máquina de Turing y Computabilidad (Cont.)

Teorema

Para cada máquina de Turing no determinista T_{nd} podemos construir una máquina de Turing determinista T_d equivalente.

>>> Tesis Turing-Church

Tesis 1 (de Turing) Un proceso naturalmente llamado procedimiento efectivo puede ser realizado por una máquina de Turing, es decir es Turing Computable.

Tesis 2 (de Church) Los procesos naturalmente llamados procedimientos efectivos o las funciones efectivamente computables son identificados con la clase de Funciones Recursivas Parciales.

>>> Tesis Turing-Church (Cont.)

1. Máquinas de Alan Turing (1936).
2. Teoría de Funciones Recursivas Parciales de K. Gödel y S. Kleene (1936).
3. λ-Cálculo de Alonzo Church (1941).
4. Sistemas Canónicos de Post (1943).
5. Algoritmo Generalizado de Markov (1951).
6. Gramáticas estructuradas por frases de Noam Chomsky (1959-1963).
7. Máquinas de registros de Shepardson y Sturgis (1963).
8. Redes de Petri (1962)

>>> Limitaciones de la Computabilidad Efectiva

Definición

Denominaremos Problema de Decisión a aquellos formulados a traves de una pregunta y que requieren una respuesta de tipo Si/No.

Definición

Un problema de decisión se dice Soluble si existe un algoritmo total para determinar si la propiedad es verdadera.

Definición

Un problema de decisión se dice Parcialmente Soluble si existe un procedimiento efectivo para determinar si la propiedad es verdadera.

Definición

Un problema de decisión es Insoluble si no existe un procedimiento efectivo para determinar si la propiedad es válida.

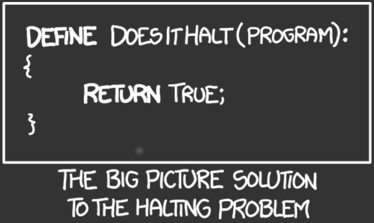
>>> Problema de la detención de la MT

Definición

Sean una máquina de Turing T y una cadena α, el problema de la detención de las máquinas de Turing se define como: ¿Existe un algoritmo para decidir si T se detendrá comenzando en el estado inicial con α en la cinta?

Teorema

El problema de la detención es Algorítmicamente Insoluble.



Demostración

Definición

Sean dos problemas de decisión, PD_1 y PD_2, diremos que el problema de decisión PD_1 se reduce al problema de decisión PD_2 y lo escribiremos $PD_1 \rightarrow_{reduct} PD_2$ si un algoritmo para solucionar PD_2 puede ser usado para construir la solución de PD_1.

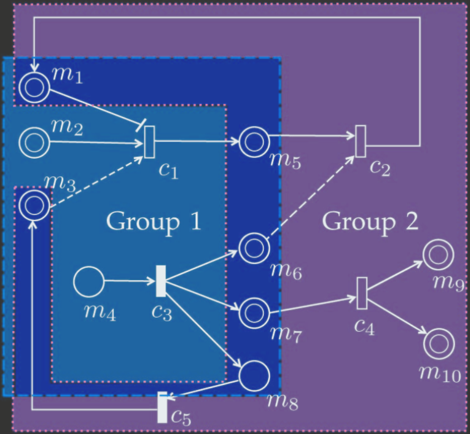
Teorema

Sean PD_1 y PD_2 dos problemas de decisión, se cumple lo siguiente:

1. Si $PD_1 \rightarrow_{reduct} PD_2$ y PD_2 es soluble, entonces PD_1 también lo es.
2. Si $PD_1 \rightarrow_{reduct} PD_2$ es insoluble, entonces PD_2 también lo es.

>>> Redes de Petri

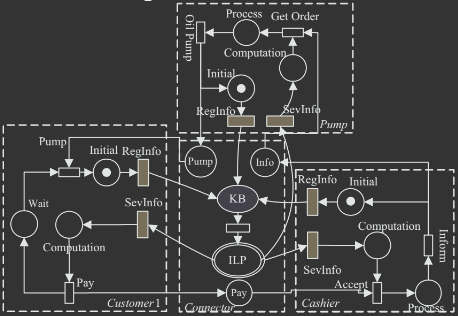
Hasta ahora hemos considerado modelos no deterministas como extensiones de otros deterministas. Las redes de Petri en cambio son inherentemente no deterministas.

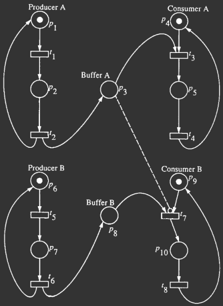


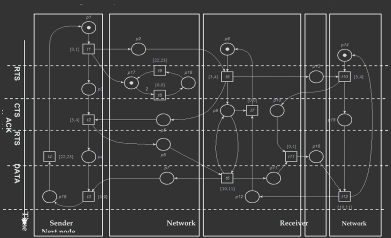
High Performance Hybrid Functional Petri Net Simulations of Biological Pathway Models on CUDA. IEEE/ACM Transactions on Computational Biology and Bioinformatics.

>>> Redes de Petri (Cont.)

Aplicaciones de Redes de Petri Protocolos de comunicación - Redes de Computadoras - Control de Procesos Industriales - Simulación - Programación Concurrente y Paralela.







>>> Redes de Petri (Cont.)

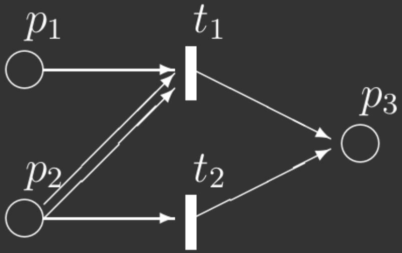
Definición

Una red de Petri (RP) es una cuaterna (P, T, IF, OF) donde:

* $P = \{p_1, p_2, \cdots, p_m\}$ es un conjunto finito de lugares.
* $T = \{t_1, t_2, \cdots, t_n\}$ es un conjunto finito de transiciones.
* $IF : P \times T \to N$ es una función de entrada a las transiciones.
* $OF : T \times P \to N$ es una función de salida de las transiciones.

>>> Redes de Petri (Cont.)

Una RP puede representarse gráficamente mediante un multidigrafo bipartito $G = (N, A)$. N es el conjunto de nodos del grafo y A el de arcos tal que $N = P \cup T$ y $P \cap T = \emptyset$.



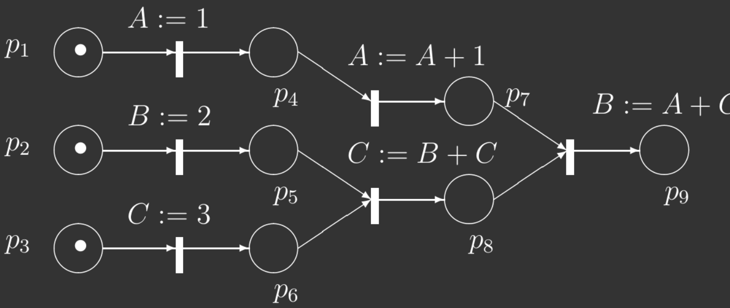
$$IF: \begin{array}{|c|cc|} \hline P \backslash T & t_1 & t_2 \\ \hline p_1 & 1 & 0 \\ p_2 & 2 & 1 \\ p_3 & 0 & 0 \\ \hline \end{array} \quad \text{y} \quad OF: \begin{array}{|c|ccc|} \hline T \backslash P & p_1 & p_2 & p_3 \\ \hline t_1 & 0 & 0 & 1 \\ t_2 & 0 & 0 & 1 \\ \hline \end{array}$$

>>> Redes de Petri (Cont.)

Definición

Un marcado de una red de Petri es un mapeo $M : P \to \mathbb{N}$.

```
A := 1;
B := 2;
C := 3;
A := A + 1;
C := B + C;
B := A + C;
```



Definición

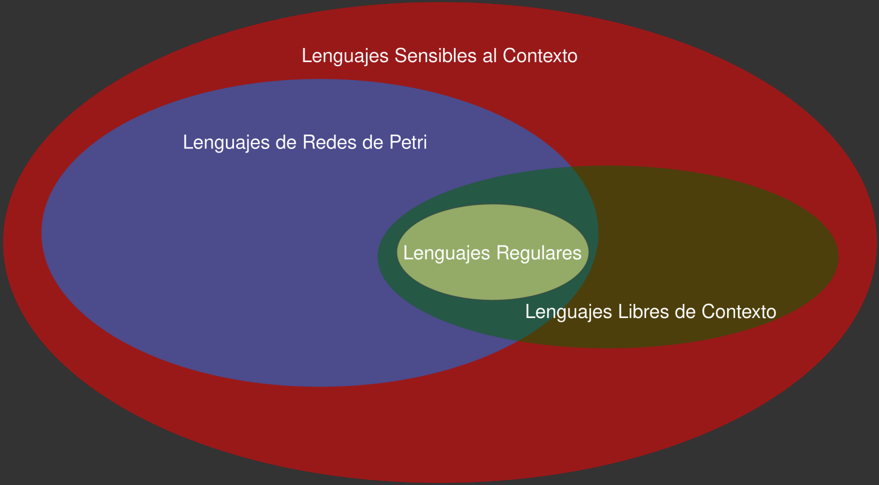
Para un dado marcado M de una red de Petri y una transición $t \in T$, diremos que t está habilitada por M si y solo si para cada lugar p, $M(p) \geq IF(p,t)$.

Definición

Sean M y M' dos marcados de una red de Petri, podemos establecer que M puede evolucionar a M', $M \vdash M'$, si y solo si existe una transición t habilitada por M tal que para cada lugar $p, M'(p) = M(p) - IF(p,t) + OF(t,p)$. A la transición seleccionada para evolucionar de M a M' se la denomina transición disparada y el disparo consiste en el cambio de marcado.

>>> Redes de Petri (Cont.)

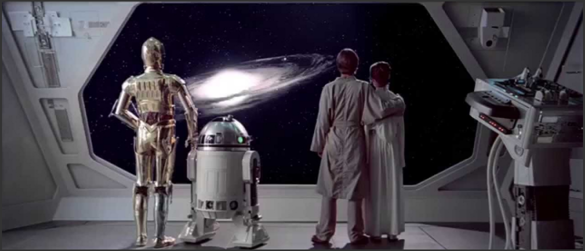
Expresividad de Redes de Petri



>>> Demo Time!

Redes de Petri en Python - Ver ipynb

\>>> Gracias!



Bibliografía

1. Petri Nets: Properties, Analysis and Applications. Tadao Murata.
2. Petri Nets Applications. Edited by Pawel Pawlewski, 2010 under CC BY-NC-SA 3.0 license DOI: 10.5772/56888
3. Introduction to Automata Theory, Languages, and Computation - Hopcroft et. al 2007 (3er ed.)
4. Teoría de la Computación - Gonzalo Navarro 2011.
5. Fundamentos de Cs. de la Computación - Juan Carlos Augusto 1995.