Universidad de Huelva

Tema 1

Introducción a los Modelos de Computación

Índice

- Computar

  Se define ***computar*** como realizar un cálculo matemático por medio de un conjunto finito de operaciones elementales.

  A finales del s:XIX los matemáticos comenzaron a plantearse no solo "qué calcular" o "qué propiedades tienen las cosas que calculamos" sino "cómo calcularlo".

  Denominamos ***algoritmo*** o ***programa*** a la secuencia de operaciones que permite realizar el cálculo.

- Modelos de Computación

  Un ***modelo de computación*** es un modelo abstracto que describe una forma de computar.

  Los aspectos a considerar al definir un modelo de computación son:

  - ¿Cómo se representan las entradas?
  - ¿Cómo se representan las salidas?
  - ¿Cuáles son las operaciones elementales?
  - ¿Cómo se combinan las operaciones para desarrollar el "programa"?

- Computabilidad

  Se denomina ***Computabilidad*** o ***Teoría de la Computación*** a la rama de la Matemática que estudia las propiedades de los modelos de computación. En particular, la Computabilidad afronta cuestiones como

  – ¿Qué funciones se pueden calcular con un determinado modelo?
  – ¿Qué funciones NO se pueden calcular con un determinado modelo?
  – ¿Qué complejidad computacional (en tiempo o en espacio) tiene la solución de un determinado problema en un determinado modelo?
  – ¿Qué tipos de problemas hay en función de su posible solución por medio de modelos de computación?
  – ¿Son equivalentes dos modelos de computación?

- La teoría clásica de la Computación se centra en el cálculo de funciones definidas sobre número naturales.
- Entre los modelos de computación estudiados se encuentran:
  - Circuitos combinacionales
  - Autómatas Finitos
  - Autómatas de Pila
  - Máquinas de Turing
  - Funciones recursivas parciales
  - Máquina de Post
  - Cálculo lambda
  - Autómatas celulares
  - Computadores cuánticos

- En la actualidad, los estudios sobre Computabilidad se centran en problemas más complejos como:
  - Máquinas con mayor poder que las máquinas de Turing. Por ejemplo, una ***máquina oráculo*** que utiliza una caja negra que puede calcular una función particular que no es calculable con una máquina de Turing. La capacidad de cómputo de una máquina oráculo viene descrita por su grado de Turing.
  - La ***teoría de cómputos reales*** estudia máquinas con precisión absoluta en los números reales. Dentro de esta teoría, es posible demostrar afirmaciones interesantes, tales como «el complemento de un conjunto de Mandelbrot es solo parcialmente decidible».

- Georg Cantor (1845-1918) (precursor)

Comenzó sus estudios universitarios en 1862 en Zurich, pero al año siguiente se desplazó la Universidad de Berlín donde se especializó en matemáticas, filosofía y física. En 1872, cuando contaba con 27 años de edad, se convirtió en catedrático en la Universidad de Halle, dando inicio entonces a sus principales investigaciones.

Desarrolló la Teoría de Conjuntos y descubrió la existencias de diferentes tipos de números infinitos (transfinitos). Comenzó sus trabajos en esta área en 1874. En 1895 y 1897 publicó dos artículos en *Mathematische Annalen* presentando todos sus avances en esta materia.

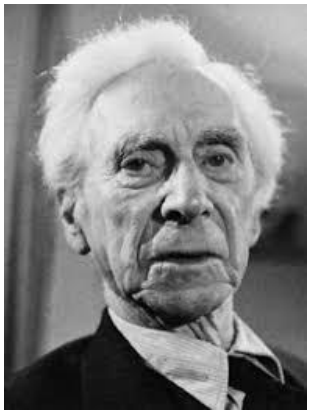
Vivió aquejado por episodios de depresión, atribuidos originalmente a las críticas recibidas y sus fallidos intentos de demostración de la hipótesis del continuo, aunque actualmente se cree que poseía algún tipo de "depresión ciclo-maníaca".

- Giussepe Peano (1858-1932) (precursor)

Entre sus múltiples aportaciones matemáticas se encuentran su desarrollo de la lógica matemática y sus aportaciones a la axiomatización de las matemáticas. Por ejemplo, los "axiomas de Peano" permiten definir completamente los número naturales:

- El 0 es un número natural.
- Si n es un número natural, entonces el sucesor de n también es un número natural.
- El 0 no es el sucesor de algún número natural.
- Si hay dos números naturales n y m con el mismo sucesor, entonces n y m son el mismo número natural.
- Si el 0 pertenece a un conjunto, y dado un número natural cualquiera, el sucesor de ese número también pertenece a ese conjunto, entonces todos los números naturales pertenecen a ese conjunto.

- Bertrand Russell (1872-1970) (precursor)



En 1890, Russell ingresó al Trinity College de Cambridge para estudiar matemáticas. En 1900 participó en el primer Congreso Internacional de Filosofía en París, donde se familiarizó con el trabajo del matemático italiano Giuseppe Peano. Se convirtió en un experto del nuevo simbolismo de Peano y su conjunto de axiomas para la aritmética.

Durante los siguientes años fue el máximo defensor del logicismo, (toda la matemática es reducible a la lógica). Entre 1910 y 1913 escribió, junto a su ex-profesor Alfred North Whitehead, la monumental Principia Mathematica, un sistema axiomático en el cual todas matemáticas pueden ser fundadas. Su plan original de incorporar la geometría en un cuarto volumen nunca fue llevada a cabo. Al completar Principia Mathematica, Russell estaba exhausto, y nunca sintió recuperar completamente sus facultades intelectuales de tal esfuerzo realizado.

- David Hilbert (1862-1943) (precursor)

Probablemente el matemático más importante de finales del s:XIX y principios del s:XX.

Desde su influencia sobre los matemáticos de la época propuso un conjunto de problemas que debían centrar el desarrollo de la matemática de su tiempo.

Entre estos estaba el problema de la decisión (*entscheidungsproblem*) que pretendía encontrar un algoritmo general que decidiera si una fórmula del cálculo de primer orden es un teorema.

En 1936, de manera independiente, Alonzo Church y Alan Turing demostraron ambos que es imposible escribir tal algoritmo.

- Kurt Gödel (1908-1978) (precursor)

Conocido por su teorema de la incompletitud (1931), que establece que para todo sistema axiomático recursivo auto-consistente lo suficientemente poderoso como para describir la aritmética de los números naturales (la aritmética de Peano), existen proposiciones verdaderas sobre los naturales que no pueden demostrarse a partir de los axiomas.

Para demostrar este teorema desarrolló una técnica denominada ahora como numeración de Gödel, la cual codifica expresiones formales como números naturales.

La demostración de Gödel fue la base de los trabajos de Church y Turing sobre el problema de la decisión.

- Moses Ilyich Schönfinkel (1889-1942)

Matemático ruso, formado en la Universidad de Odessa. Entre 1914 y 1924 formó parte del grupo de David Hilbert en la Universidad de Göttingen.

En 1920, en una charla interna del grupo presentó las bases de la Lógica combinatoria. Este trabajo se publicó en 1924. En 1929 se publicó un segundo trabajo sobre el problema de la decisión.

Tras dejar Göttingen en 1924 volvió a Moscú. En 1927 sufrió una enfermedad mental. El resto de su vida lo pasó en una pobreza absoluta. No se conserva nada de sus trabajos ya que sus papeles fueron quemados por sus vecinos para calentarse.

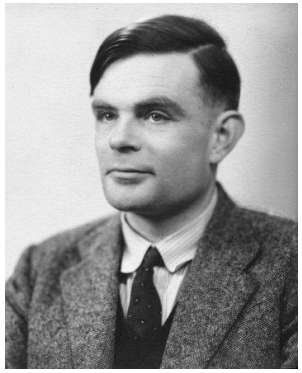
- Alonzo Church (1908-1978)

Su obra más conocida es el desarrollo del cálculo lambda, y su trabajo de 1936 que muestra la existencia de problemas indecidibles.

Demostró (junto a Turing) que el cálculo lambda y las máquinas de Turing son equivalentes. Posteriormente demostraron que una variedad de procesos mecánicos alternos para realizar cálculos tenían poder de cómputo equivalente. Como resultado se postuló la Tesis de Church-Turing.

Entre los más conocidos estudiantes de doctorado de Church están Stephen Kleene, J. Barkley Rosser, Leon Henkin, John George Kemeny, Michael O. Rabin, Dana Scott, Simon Kochen, Raymond Smullyan y otros.

- Alan Turing (1912-1954)



En su trabajo "Los números computables, con una aplicación al Entscheidungsproblem" (publicado en 1936), Turing reformuló los resultados obtenidos por Kurt Gödel en 1931 sobre los límites de la demostrabilidad y la computación, sustituyendo al lenguaje formal universal descrito por Gödel por lo que hoy se conoce como Máquina de Turing, unos dispositivos formales y simples. Demostró que dicha máquina era capaz de implementar cualquier problema matemático que pudiera representarse mediante un algoritmo.

- Haskell B. Curry (1900-1982)

Matemático estadounidense. Estudió en la Universidad de Harvard (comenzó Medicina antes de decidirse por Matemáticas), en el MIT (Ingeniería Eléctrica) y de nuevo en Harvard (Física). En esta época comienza a interesarse por la lógica simbólica.

En 1927 descubre que su trabajo en Lógica Combinatoria ya había sido desarrollado antes por Moses Schönfinkel y se desplaza a Göttingen para trabajar en esta línea dentro del grupo de Hilbert, consiguiendo el doctorado en Matemáticas en 1930.

En 1929 consigue una plaza en el Pennsylvania State College donde pasa casi toda su vida. Su trabajo principal fue el desarrollo de la Lógica Combinatoria como fundamento de las Matemáticas.

- Stephen Kleene (1909-1994)

Alumno de Church, trabajó con él desde 1930 en el desarrollo del cálculo lambda. En 1934 obtuvo el título de doctor por la tesis "Una teoría de los enteros positivos en Lógica Formal", dirigida por Church.

En 1935 consigue una plaza como profesor en la Universidad de Wisconsin-Madison, donde pasa casi toda su vida.

En 1940 desarrolla la teoría de la recursión, como un modelo de computación alternativo.

En los 50s participa también en el desarrollo de la Teoría de Autómatas. Se le conoce sobre todo como creador del operador de clausura (*) .

- Stephen A. Cook (1939-)

Graduado en Matemáticas por la Universidad de Michigan (1961), Máster en Matemáticas (1965) y Doctor en Matemáticas (1966) en la Universidad de Harvard. Profesor de la Universidad de Berkeley entre 1966 y 1970 y de la Universidad de Toronto desde entonces.

Se le considera uno de los padres de la Teoría de la Complejidad Computacional. En 1971 publica el artículo "The Complexity of Theorem Proving Procedures" en el que formaliza los conceptos de reducción en tiempo polinomial, NP-completitud, prueba la existencia de un problema NP-completo y formula el problema P vs NP.

- Leonid Levin (1948-)

Máster en Matemáticas (1970) en la Universidad de Moscú. En 1978 se desplazó a EEUU y logró el Doctorado en Matemáticas por el MIT en 1979. Desde 1980 es profesor en la Universidad de Boston.

Demostró la existencia de problemas NP-completos de forma independiente a Cook (lo que se conoce como teorema de Cook-Levin). Publicó sus resultados en 1973 (después de Cook) aunque presentó estos resultados en varios seminarios anteriores.

- Teoría de conjuntos
  - Un **conjunto** es una colección de elementos desordenada y sin repetición.
  - Operaciones básicas entre conjuntos: **unión**, **intersección** y **diferencia**.
  - Dados dos conjuntos, *A* y *B*, se dice que *A* es **subconjunto** de *B* si todos los elementos de *A* pertenecen a *B*.

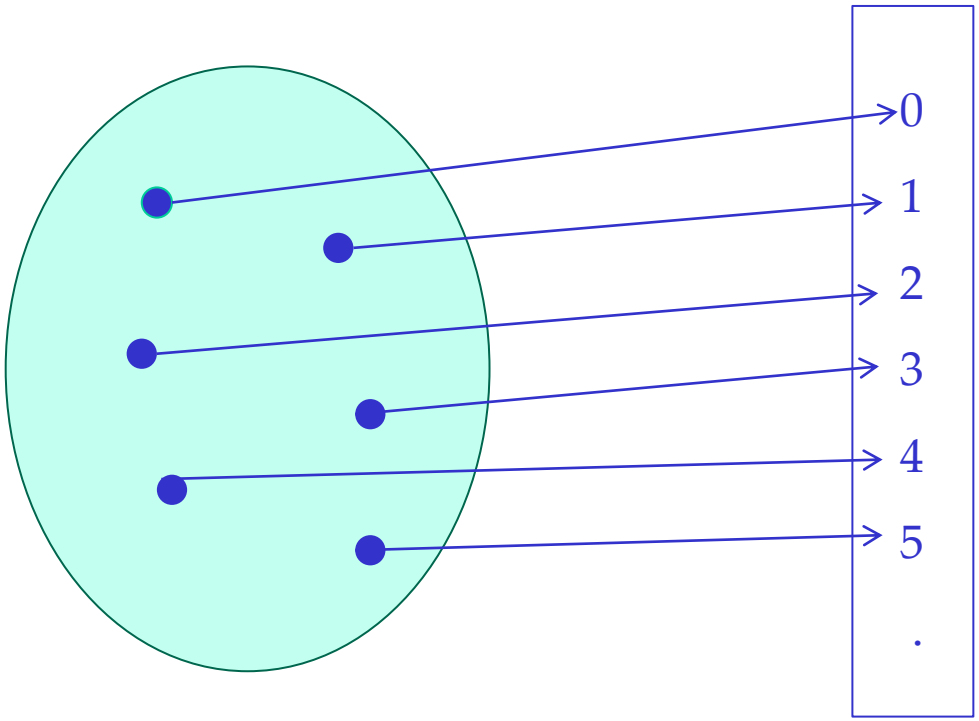
- Teoría de conjuntos
  - Dados dos conjuntos, *A* y *B*, se denomina ***Producto Cartesiano*** (*A* x *B*) al conjunto formado por pares de elementos *(a,b)* en los que el primer elemento de cada par pertenece al conjunto *A* y el segundo elemento de cada par pertenece al conjunto *B*.
  - Dados dos conjuntos, *A* y *B*, se denomina ***relación*** a un subconjunto del producto cartesiano *A*x*B*.

- Teoría de conjuntos
  - Se denomina ***función*** o ***aplicación*** ($A \rightarrow B$) a una relación en la que para cada elemento de A no existe más de un par en la relación (a,b).
  - Se denomina ***función total*** a una función en la que para cada elemento de A existe un par (a,b). En caso contrario se denomina ***función parcial.***

- Teoría de conjuntos
  - Se denomina **cardinalidad** de un conjunto al número de elementos que contiene.
  - La cardinalidad puede ser finita o infinita (**números transfinitos**).
  - Dado un conjunto *A* se denomina **potencia de A** al conjunto formado por todos los subconjuntos de *A*. La cardinalidad del conjunto potencia es $2^{|A|}$.
  - La cardinalidad del conjunto de los números naturales se denomina $\aleph_0$ (**aleph-0**).
  - Se pueden construir conjuntos con cardinalidades superiores. Se denomina $\aleph_1$(**aleph-1**) al cardinal inmediatamente superior a $\aleph_0$.

- Teoría de conjuntos
  - Se dice que un conjunto *A* es ***numerable*** o ***contable*** si es posible construir una correspondencia biunívoca entre el conjunto *A* y el conjunto de los números naturales o un subconjunto de los números naturales.

- Conjuntos contables
  - Conjuntos finitos



- Conjuntos contables
  - Números enteros $\mathbb{Z}$

Ejemplo de correspondencia ($\mathbb{Z} \rightarrow \mathbb{N}$)

- Si z<0 entonces n = 2·z + 1
- Si z>=0 entonces n = 2·z

- Conjuntos contables
  - Parejas de números naturales $(\mathbb{N}, \mathbb{N})$

Ejemplo de correspondencia $(\mathbb{N}, \mathbb{N}) \rightarrow \mathbb{N}$

- $n = b + (a+b) \cdot (a+b+1) / 2$

$$\begin{bmatrix} 0 & 2 & 5 & 9 & .. \\ 1 & 4 & 8 & .. & .. \\ 3 & 7 & .. & .. & .. \\ 6 & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \end{bmatrix}$$

- Conjuntos contables
  - Números racionales $\mathbb{Q}$
  - Se puede entender como $\mathbb{N}\times\mathbb{N}$, es decir, parejas de números naturales (numerador, denominador)
  - Hay que tener en cuenta las equivalencias (p.e. , 1/2 = 2/4)
  - En realidad, cada número racional representa una clase de equivalencia en $\mathbb{N}\times\mathbb{N}$.
  - Por tanto, $card(\mathbb{N}) \leq card(\mathbb{Q}) \leq card(\mathbb{N}\times\mathbb{N}) = card(\mathbb{N})$,
  - es decir, $card(\mathbb{Q}) = card(\mathbb{N}) = \aleph_0$

- Conjuntos contables
  - Conjuntos finitos de números naturales {a,b,c,d}
  - Se pueden codificar como potencias de números primos

$$\{a, b, c, d\} \rightarrow 2^a \cdot 3^b \cdot 5^c \cdot 7^d$$

  - Esta relación permite codificar listas ordenadas de números naturales (no solo conjuntos).

- Teorema de Cantor
  - *"El conjunto potencia de cualquier conjunto A tiene una cardinalidad estrictamente mayor que la cardinalidad del propio A."*

Demostración.

- Sea f: $A \rightarrow$ Pot(A). El Teorema de Cantor indica que la función no es sobreyectiva, es decir, que hay elementos de Pot(A) que no tienen un elemento en A.
- Sea B $\in$ Pot(A) el subconjunto definido como

$$B = \{x \in A : x \notin f(x)\}$$

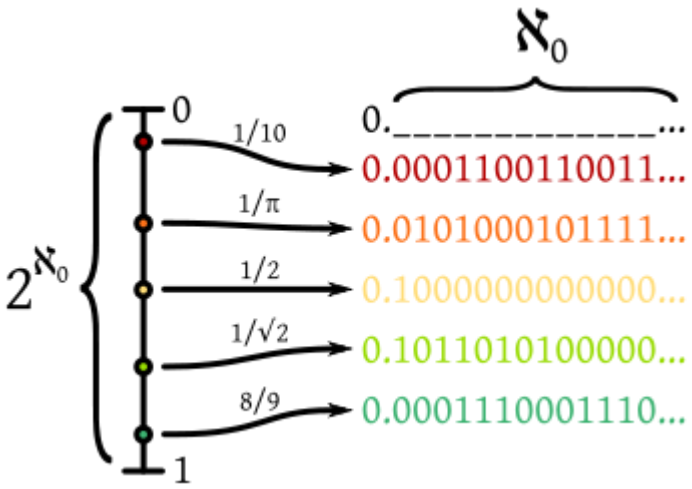
- Si el Teorema de Cantor es falso, entonces existe un x tal que $f(x) = B$. Pero entonces, si $x \in B \Longrightarrow x \notin f(x)$ y si $x \notin B \Longrightarrow x \in f(x)$. Como hemos dicho que $f(x) = B$ esto es una contradicción.

- Conjuntos incontables
  - Conjuntos de números naturales (finitos o infinitos)
  - Se trata del conjunto potencia de $\mathbb{N}$.
  - Según el ***teorema de Cantor***, card(A) < card(potencia(A))
  - La cardinalidad del conjunto potencia($\mathbb{N}$) es $2^{\aleph_0}$, es decir,

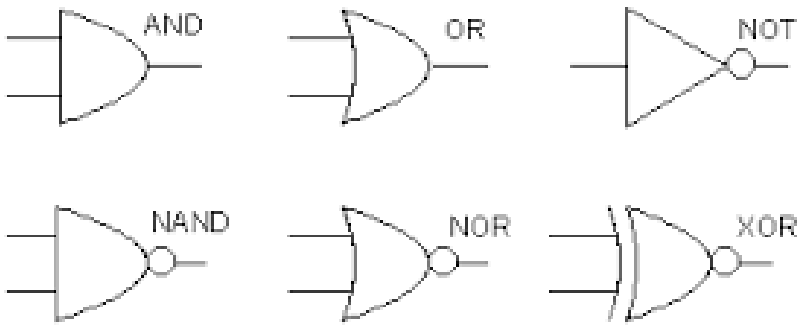
    $$\aleph_1 \leq \text{card(potencia(} \mathbb{N} \text{))}$$

  - La ***hipótesis del continuo*** establece que $\aleph_1$ = card(potencia($\mathbb{N}$)), pero se trata de una hipótesis. Existen teorías de conjuntos que no cumplen esta hipótesis.

- Conjuntos incontables
  - Número reales $\mathbb{R}$
  - La cardinalidad de $\mathbb{R}$ es igual a la de potencia($\mathbb{N}$).



- Circuitos combinacionales
  - Entradas: codificación binaria.
  - Salidad: codificación binaria
  - Operaciones elementales: puertas lógicas



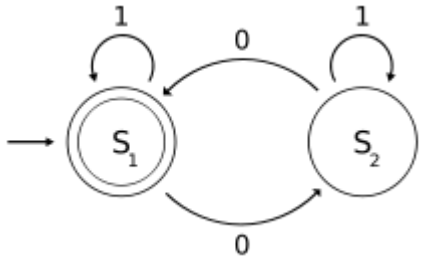
  - Reglas de composición: conexión entre circuitos formando grafos dirigidos acíclicos

- Autómatas finitos

  - Entradas: cinta sin fin formada por celdas que almacenan símbolos (codificaciones binarias, decimales u otras)
  - Salidas: cinta sin fin formada por celdas que almacenan símbolos
  - Operaciones elementales: transición de estado, leyendo un símbolo de la cinta de entrada y escribiendo un símbolo en la cadena de salida.

    (q0, a; q1, b)

  - Regla de composición: Añadir estados y transiciones.



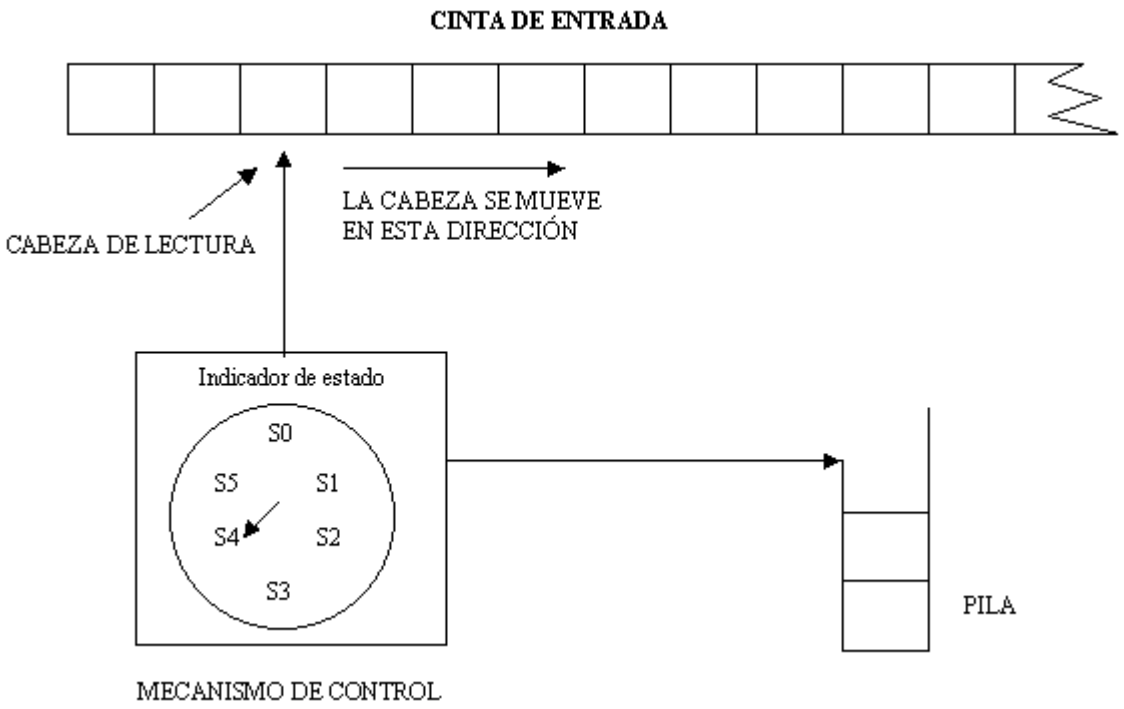
- Autómatas de pila

  - Entradas: cinta sin fin formada por celdas que almacenan símbolos (codificaciones binarias, decimales u otras)
  - Salidas: cinta sin fin formada por celdas que almacenan símbolos
  - Pila de símbolos (operaciones push y pop)
  - Operaciones elementales: transición de estado (q0 a q1), leyendo un símbolo de la cinta de entrada (a) y otro de la pila (m) y escribiendo un símbolo en la cadena de salida (b) y en la pila (n).

    (q0, a, m; q1, b, n)

  - Regla de composición: Añadir estados y transiciones.

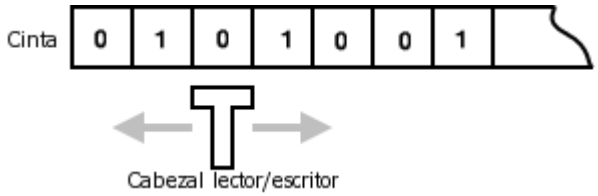
- Autómatas de pila



- Máquinas de Turing

  - Entradas: cinta sin fin formada por celdas que almacenan símbolos
  - Salidas: contenido final de la cinta
  - Operaciones elementales: transición de estado (q0 a q1), leyendo un símbolo de la cinta (a), escribiendo un símbolo en la cinta (b) y realizando un movimiento sobre la cinta (izquierda o derecha).

    (q0, a; q1, b, L/R)
  - Regla de composición: Añadir estados y transiciones.



- Máquinas de Post
  - Propuesta por Emil Post en 1936 (de forma independiente a Turing)
  - Entradas: cinta sin fín en los dos sentidos formadas por celdas que pueden estar marcadas o sin marcar (1 o 0).
  - Salidas: contenido final de la cinta
  - Operaciones elementales:
    - Marcar celda (WRITE 1)
    - Desmarcar celda (WRITE 0)
    - Mover a la derecha (MOVE RIGHT)
    - Mover e la izquierda (MOVE LEFT)
    - Leer cabezal, si está marcado ejecutar instrucción i-ésima, sino ejecutar instrucción j-ésima (IF … THEN … ELSE …)

- Máquinas de acceso aleatorio (RAM)
  - Formada por una memoria de acceso aleatorio, un conjunto de registros y un programa (secuencia de instrucciones).
  - Entradas: contenido almacenado en los registros de entrada
  - Salidas: contenido almacenado en los registros de salida
  - Se permite direccionamiento directo o indirecto
  - Operaciones básicas (instrucciones base):
    - CLR Ri : limpia el registro.
    - INC Ri: incrementa el contenido del registro.
    - JMP0 Ri Lj : si el contenido de Ri es 0 salta a la instrucción etiquetada Lj, si no continua secuencialmente.
    - HALT: finaliza la ejecución. El valor de salida será el almacenado en los registros de salida.

- Funciones Recursivas Parciales (Gödel-Kleene)
  - Funciones ($\mathbb{N} \times \mathbb{N} \times \ldots) \rightarrow \mathbb{N}$
  - Funciones base:
    - Función cero: 0(x) -> 0
    - Función sucesor: S(x) -> x+1
    - Función proyección: $U_i^n(x_1, \ldots, x_n) = x_i$
  - Operaciones:
    - Sustitución (composición): $h(\mathbf{x}) = f(g(x))$;
    - Recursión: $h(x,0) = f(x)$; $h(x,y+1) = g(x,y, h(x,y))$;
    - Minimalización: $\mu_y(\mathbf{x}) = \min y \mid f(\mathbf{x},y) = 0$

- Cálculo λ (Church)
  - Término:
    - Toda variable es un término: x, y, z,
    - Si *t* es un término y *x* es una variable, (λx. t) es un término llamado ***abstracción lambda***. Forma una función que liga la variable x con el término t.
    - Si t y s son términos, (t s) es un término llamado ***aplicación lambda***.
  - Se pueden definir los números naturales como
    - 0:= λ f x. x
    - 1:= λ f x. f x
    - 2:= λ f x. f (f x)

- Lógica combinatoria (Schönfinkel , Curry)
  - Se basa en generar funciones mediante la combinación de funciones primitivas
  - Términos combinatorios:
    - Toda variable es un término: x, y, z,
    - Toda función primitiva es un término
    - Si t y s son términos, (t s) es un término.
  - Combinadores
    - (**I** x) = x (identidad)
    - ((**K** x) y) = x (funciones constantes)
    - (**S** x y z) = ((x z) (y z))
    - (C a b c) = (a c b)
    - (B a b c) = (a (b c))

- Lógica combinatoria (Schönfinkel , Curry)
  - El combinador I se puede obtener mediante S y K.
  - Los combinadores S y K (propuestos por Schönfinkel) forman una base completa y se puede demostrar que todo término lambda puede expresarse como término combinatorio utilizando estas funciones primitivas.
  - Los combinadores B y C (propuestos por Curry) también forman una base completa.