

En la teoría de la computación, la relación entre las máquinas de Turing no determinísticas (MTN) y las máquinas de Turing determinísticas (MTD) es comparable a la relación, en el marco del desarrollo de software, entre la especificación y la construcción de programas, es decir entre la descripción de qué debe hacer y cómo debe comportarse un artefacto de software.

Se define que una MTN acepta una cadena si al menos alguna de sus computaciones la acepta; no importa cómo se implemente, una MTN acepta una cadena w si en al menos una de sus computaciones se detiene en el estado q_A a partir de w . Dos implementaciones posibles de una MTN, a ser mencionadas sólo a los fines de entender el concepto del no determinismo y no con sentido práctico, son la ejecución secuencial o en paralelo de todas sus computaciones (esta última implementación puede llegar a involucrar el uso de un número excesivamente grande de “procesadores”, no factible en la práctica).

Se debe tomar a una MTN como una adecuada especificación de una MTD (contraposición de máquina ideal vs. máquina real). La descripción de la solución de un problema se simplifica por medio del uso de MTN, con respecto al uso de MTD. Sea por ejemplo el problema de la unión de dos lenguajes recursivamente numerables L_1 y L_2 . Si δ_1 y δ_2 son las funciones de transición de las MT que reconocen L_1 y L_2 , respectivamente (suponiendo que dichas máquinas son determinísticas), una solución trivial al problema es construir una MTN M con una relación (ya no una función) de transición Δ que incluya a todas las tuplas de $\delta_1 \cup \delta_2$, y agregue dos nuevas tuplas, permitiendo como primer paso de M la bifurcación no determinística a la primera configuración de M_1 (la máquina que acepta L_1) o a la primera configuración de M_2 (la máquina que acepta L_2). Como se indicó antes, podría considerarse a M como una especificación del algoritmo que resuelve el problema (por ejemplo, la ejecución “en paralelo” de M_1 y M_2).

En otras disciplinas, como por ejemplo la verificación de programas, no es habitual considerar el enfoque del no determinismo de la teoría de la computación: ahora, que un programa no determinístico S termine satisfaciendo una propiedad p , significa que

Máquinas de Turing no determinísticas

p se cumpla al final de toda computación de S (se prioriza la eficiencia del compilador o intérprete).

Las MTN son fundamentalmente útiles para el análisis de la complejidad computacional de los problemas. En el caso específico de la complejidad computacional temporal, un vasto conjunto de problemas de interés informático (de la teoría de grafos, la lógica, la aritmética, etc.) tienen resoluciones no determinísticas polinomiales (y no se les han encontrado resoluciones determinísticas polinomiales). Se denomina NP al conjunto de dichos problemas (por su parte P es el conjunto de los problemas que se pueden resolver en tiempo determinístico polinomial, que por convención – consistente con la práctica cotidiana – se considera eficiente). Uno de los interrogantes más importantes en este marco es si NP incluye estrictamente a P (por la relevancia de los problemas de NP). Se sospecha que sí, pero aún no se ha podido probar formalmente.