Un lenguaje reconocible pero no decidible Ya que las máquinas de Turing pueden ser codificadas como palabras finitas, entonces una máquina de Turing w podría ser el input de otra TM. Además, el par máquina de Turing M, palabra w, también podría ser el input de una TM. Por el resto de esta asignatura, supondremos que si $< M, w >$ (la codificación de M y w) es el input para una TM H, entonces H siempre puede simular lo que haría M cuando el input es w.

Definimos el siguiente lenguaje:

$$A_{TM} = \{\langle M, w\rangle : \ M \text{ acepta } w\}$$

Teorema 2.22. *El lenguaje A_{TM} es reconocible pero no es decidible.*

Proof. Lo primero es ver que A_{TM} es reconocible. Para ver esto basta con, para cada entrada $\langle M, w\rangle$, simular lo que hace M cuando su entrada es w. Como estamos suponiendo que eso se puede hacer siempre, entonces A_{TM} es reconocible.

Ahora, supongamos que A_{TM} es decidible. Es decir, supongamos que existe una TM H que para cada entrada $\langle M, w\rangle$ se detiene, acepta cuando M acepta a w y rechaza cuando M no acepta a w.

Ahora construiremos una nueva TM D a partir de H. Esta nueva TM D ejecuta H como subrutina, entonces, para cada entrada $\langle M, w\rangle$ usa a H para determinar qué hace M cuando su entrada es $w = \langle M\rangle$, una codificación de la misma TM M. Una vez que D ha determinado dicha información, D hará lo opuesto. Es decir, rechaza si M acepta a $\langle M\rangle$ y acepta si M rechaza a $\langle M\rangle$.

Ahora veremos lo que ocurre cuando D recibe como input a $\langle D\rangle$, una codificación de sí misma. Bueno, si H dice que D acepta $\langle D\rangle$, entonces D rechaza a $\langle D\rangle$, y si H dice que D rechaza a $\langle D\rangle$, entonces D acepta a $\langle D\rangle$. Esto es claramente una contradicción, por lo tanto D no puede existir, y luego H no puede existir. Lo que nos permite concluir que A_{TM} no es decidible. □

Halt: Definimos el siguiente lenguaje.

$$HALT_{TM} = \{\langle M, w\rangle : \ M \text{ se detiene con la entrada } w\}$$

Teorema 2.23. *El lenguaje $HALT_{TM}$ no es decidible.*

Proof. Supongamos que $HALT_{TM}$ es decidible. Es decir, supongamos que existe una TM R que acepta cada vez que para una entrada $\langle M, w\rangle$, M se detiene cuando la entrada es w, y rechaza cuando M no se detiene cuando la entrada es w.

Ahora construiremos una nueva TM S a partir de R que decide A_{TM}. Como sabemos que A_{TM} no es decidible, entonces tal TM R no puede existir. Esta nueva TM S ejecuta R como subrutina, entonces, para cada entrada $\langle M, w\rangle$ usa a R para determinar si M se detiene cuando su entrada es w. Una vez que R ha determinado dicha información, S hará lo siguiente:

- Si R rechaza, entonces S rechaza (en este caso, R nos dice que M no se detiene cuando el input es w).
- Si R acepta, entonces S simula M con entrada w hasta que se detenga (R garantizó que M se detendrá cuando la entrada es w). Si M acepta, S acepta, y si M rechaza entonces S rechaza.

$\square$

De esta manera, hemos visto que existen lenguajes no decidibles pero reconocibles, y lenguajes no reconocibles.