MT2 - grades + stats today

HW 11 due today.

GPS11 due Mon Dec 8th

HW12 "due" Mon 8th.

FLEX forms

CA applications

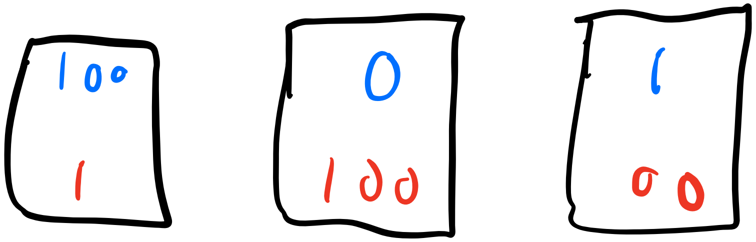
Fix an alphabet ($\Sigma = \{0,1\}$). Given a set of dominoes. Each has a top & bottom string.

$$\left[\frac{0}{100}\right] \quad \left[\frac{01}{00}\right] \quad \left[\frac{110}{11}\right]$$

Want to build a sequence of dominoes. Can repeat them an arbitrary # times.

Can we make a sequence where concat of top strings = concat of bottom strings?

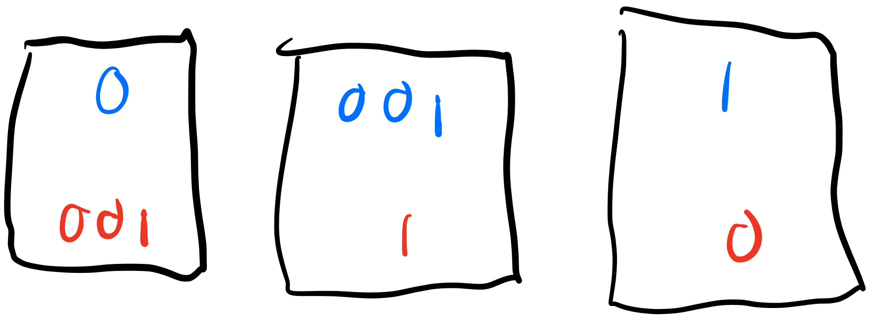
$$\left[\frac{110}{11}\right] \left[\frac{01}{00}\right] \left[\frac{110}{11}\right] \left[\frac{0}{100}\right] \leftarrow \text{Yes!}$$



← No!

Post Correspondence Problem [Post '46]

Can find yes answers by enumerating all sequences.

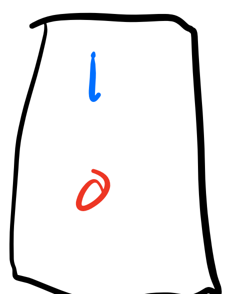


← Yes, with 75 dominoes.

No bound on max length.

No algorithm to guarantee answer is no.

"undecidable"

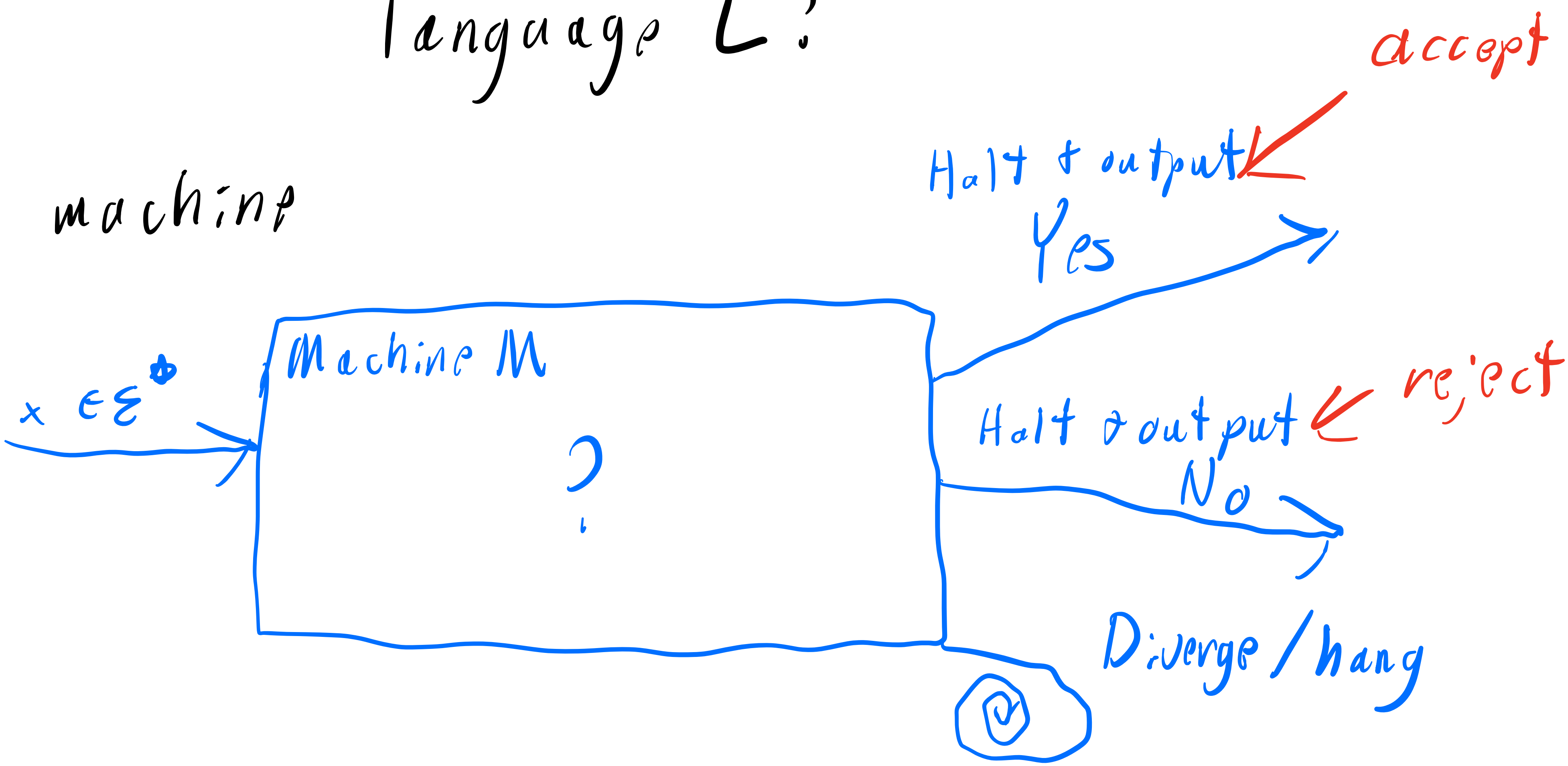


"Machine": model of arbitrary computation

decision problem: input is a string $x \in \Sigma^*$
want to return Yes or No

i.e. does x belong to a certain language L?

A machine



Given machine M, four languages:

$$\text{Accept}(M) := \{x \in \Sigma^* \mid M \text{ accepts } x\}$$

$$\text{Reject}(M) := \{x \in \Sigma^* \mid M \text{ rejects } x\}$$

$$\text{Halt}(M) := \text{Accept}(M) \cup \text{Reject}(M)$$

$$\text{Diverge}(M) := \Sigma^* \setminus \text{Halt}(M)$$

(Hang)

Language L is <u>acceptable</u> if there exists at least one machine M s.t. $\text{Accept}(M) = L$.

(applies to corresponding decision problem)

(Post Cor. Problem)

Language L is <u>decidable</u> if there exists a machine M such that $L = \text{Accept}(M)$ &

$\text{Diverge}(M) = \emptyset$.

<u>undecidable</u>: not decidable

Thm: There is an undecidable language.

Can <u>describe</u> any machine M using a string $\langle M \rangle$.

Countably many machines $\Rightarrow$ countably many decidable languages.

But uncountably many languages!

SelfReject: $\{\langle M \rangle \mid M \text{ rejects } \langle M \rangle\}$

Thm: SelfReject is undecidable.

Proof: Suppose for contradiction there is a machine SR that decides SelfReject.

SR accepts or rejects. It does not diverge.

Consider feeding $\langle SR \rangle$ to SR.

Suppose SR accepts $\langle SR \rangle$.

$\Rightarrow$ SR rejects $\langle SR \rangle$. $\perp$

Suppose SR rejects $\langle SR \rangle$.

$\Rightarrow$ SR accepts $\langle SR \rangle$. $\perp$

Our original assumption on SR existing must be wrong. <u>diagonalization</u>

Halting problem

$$\text{Halt} := \{ \langle M, x \rangle \mid M \text{ halts on } x \}.$$

machine

input for M

```
CheckGoldbach():
    n = 4
    while True:
        flag = False
        for i from 2 to n - 2:
            if i and (n - i) are both prime:
                flag = True
        If flag is False:
            return False
        n = n + 2
```

Goldbach's conjecture:

Every even integer ≥ 4 is the sum of two primes.

If Halt is decidable, we have an algo to solve the conjecture.

- A description <M> can be executed + modified.

We can write Python interpreters in Python.

Thm: Halt is acceptable.

Proof: Given <M, x>, run M on x. If it halts, return Yes. ✓

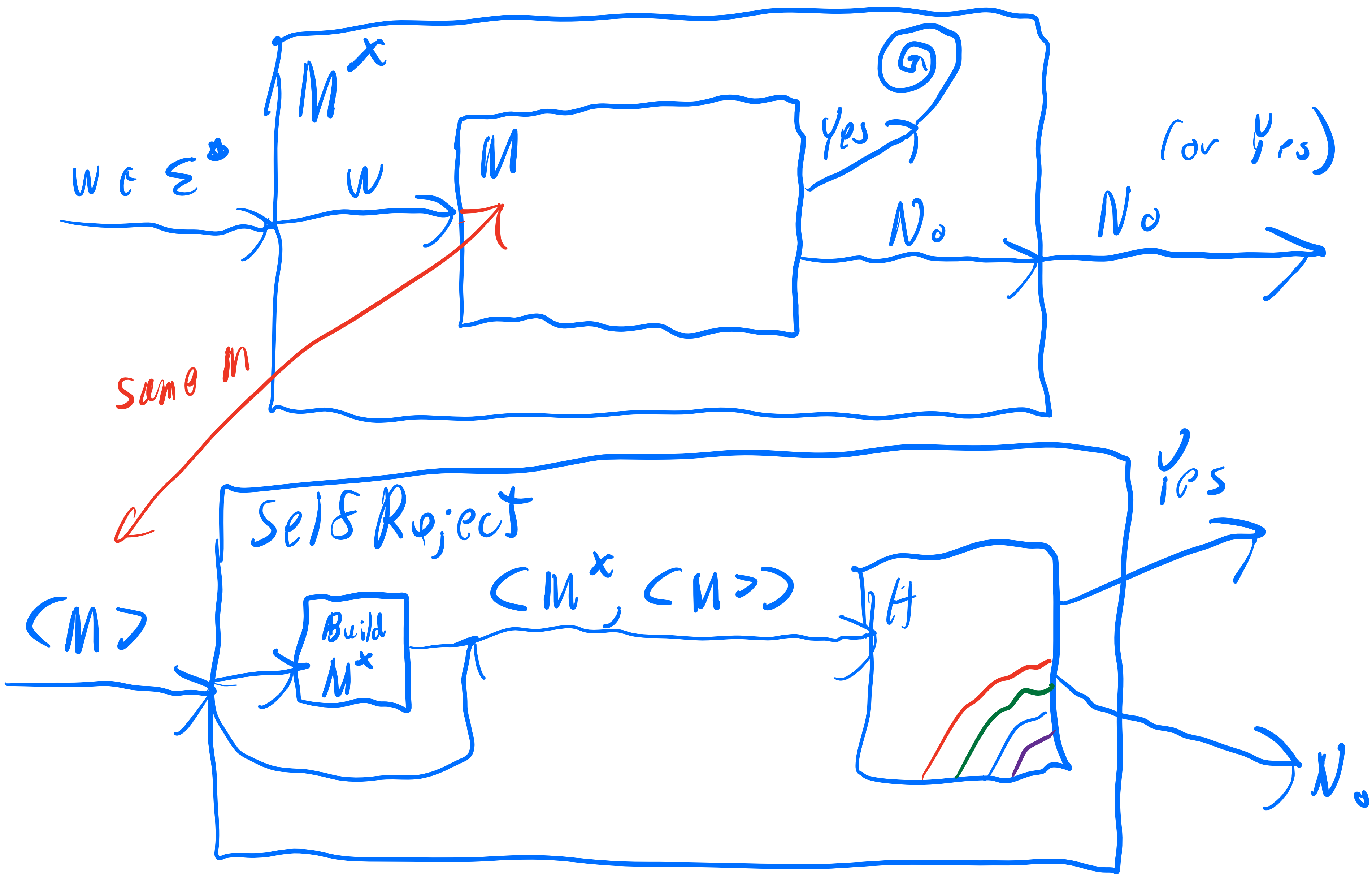
Thm: Halt is undecidable.

Proof: Reduce SelfReject to Halt.

Suppose there is a machine H to decide Halt.

Let $\langle M \rangle$ be an input to SelfReject.

Let M^x be a machine:



But SelfReject is undecidable. $\perp$

H does not exist.