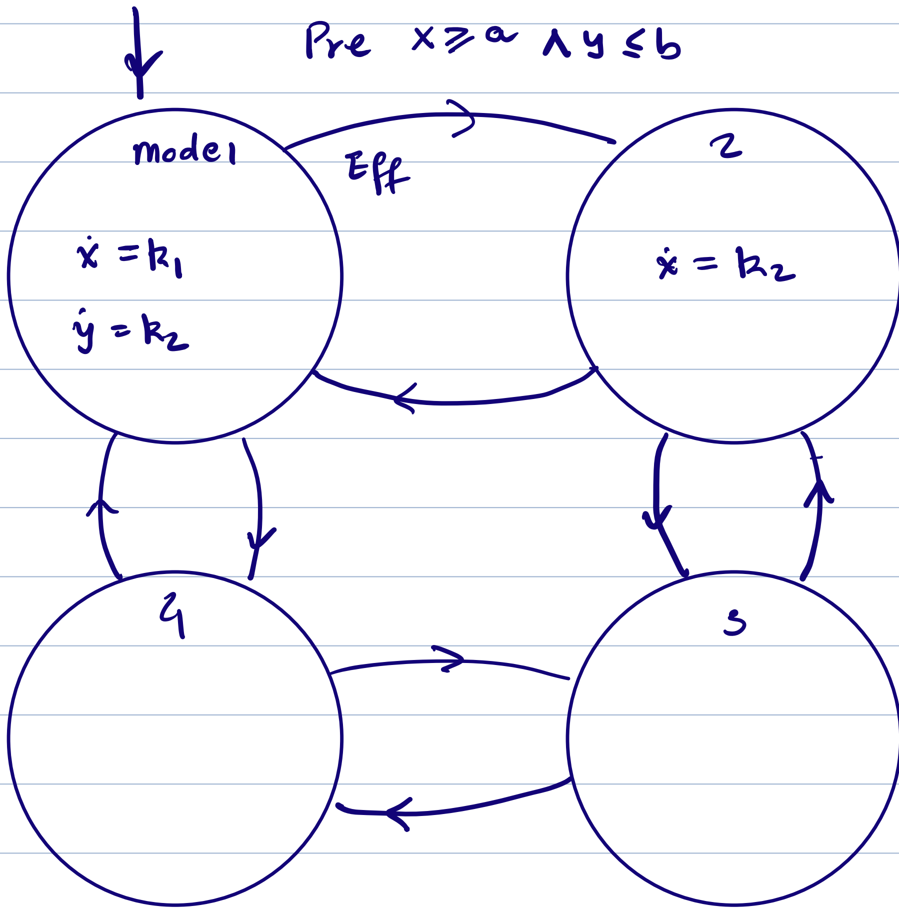
Rectangular Hybrid Automata ?

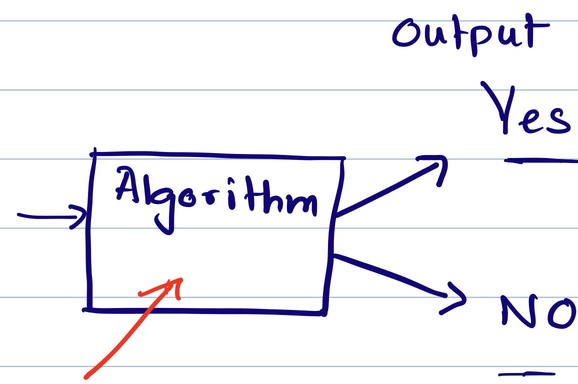


Given location $\{3\}$ Reachable?

Verification problem from last lecture

Input : A Rectangular Hybrid Automaton

$\underline{\ell} \in L$ a mode location



Can you find such an algorithm ?

Not decidable!

Henzinger, Kopke, Puri, Varaiya 1995
Whats Decidable About Hybrid Automata?

Control state reachability problem for Rectangular Hybrid automata is <u>undecidable</u>.

No clever scaling of clocks will make A a FSM

No other tricks to create any algorithm

A much faster GPU will not help

No one else can come up with any Alg.

How do you prove a result like this?

Computability theory

Decision problems. A problem where we expect yes/no answer on any input.

E.g. Is x Composite?
Is a vertex in T reachable from any vertex in S in the graph $G = \langle V, E \rangle$?

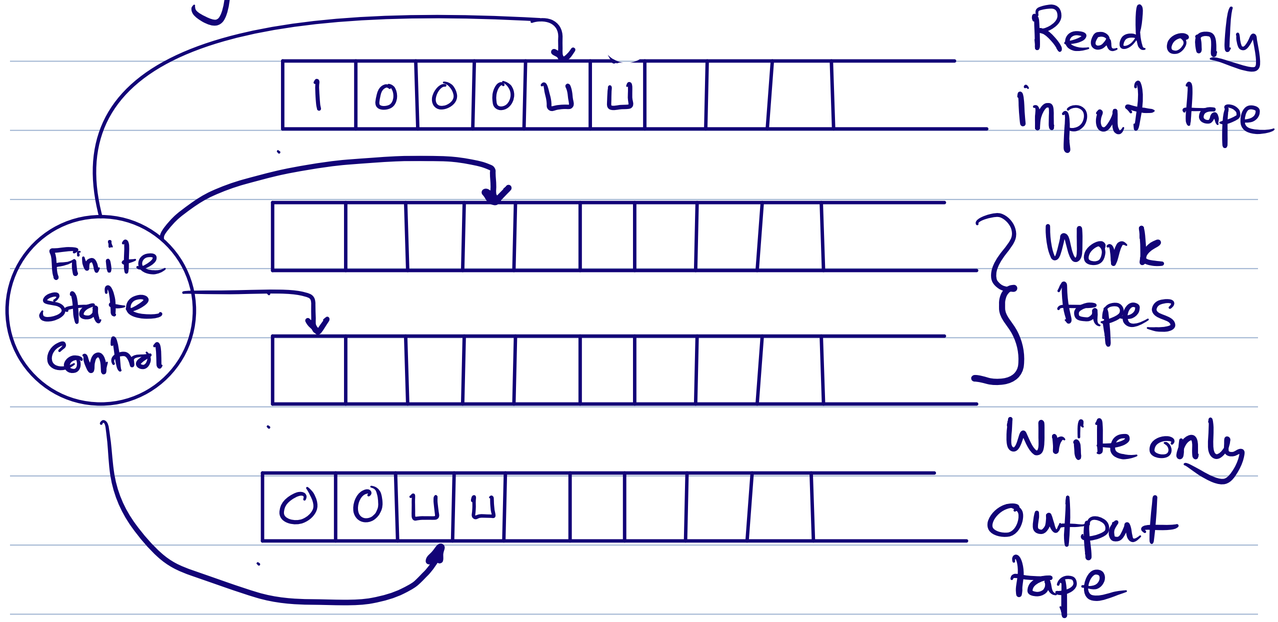
Think of Decision problems as sets of string defining a Language

Σ : alphabet e.g. $\Sigma = \{0, 1\}$
Σ^* : set of strings of 0 or finite length.
$w \in \Sigma^*$ particular string
w_i i^{th} symbol in the string
$w = 110011$

Example $L_c = \{x \in \Sigma^* \mid x \text{ is Composite}\}$
$= \{10, 100, 110, 1000, \cdots\}$

$L_{AG} = \{x \in \Sigma^* \mid x \text{ is an encoding of a graph with non-negative cycles}\}$

Turing Machines



initially

one step

Deterministic Turing machine (DTM)

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej} \rangle$

Q : Finite set of control states
Σ : Finite input alphabet
$\Gamma \supseteq \Sigma$: Finite tape symbols $\sqcup \in \Gamma / \Sigma$
$q_0 \in Q$ initial state
$q_{acc} \in Q$ accept state
$q_{rej} \in Q$ reject state $q_{rej} \neq q_{acc}$

$$\delta : (Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma^{k+1} \rightarrow$$

$$Q \times \{L, S, R\} \times (\Gamma \times \{L, S, R\})^k \times (\Gamma \cup \{\varepsilon\})$$

transition function

Example : 1-tape TM for checking $L_{pal} = \{ x x^{-1} \mid x \in \Sigma^* \}$
palindromes

Executions of a TM

State / Configuration

- Control state
- Contents of all work-tapes
- positions of all heads

$c \in Q \times \{0, \dots, n-1\} \times (\Gamma^* \{*\} \Gamma \Gamma^*)^k$

special pos of head

init:

$c_0 := (q_0, 0, *\sqcup, *\sqcup, \dots *\sqcup)$

Halting config:

$(q_{acc}, \dots)$ OR

$(q_{rej}, \dots)$

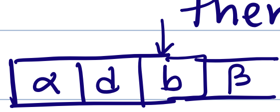
Accepting config where control state is q_{acc}

$C_1 \longrightarrow C_2$

E.g. if $\delta(q, a, b) = (q', R, (c, L))$

and input w with $w_i = a$

then $(q, i, \alpha d * b \beta) \longrightarrow$



$(q', i+1, \alpha * d c \beta)$

$C_0 \xrightarrow{w_0} C_1 \xrightarrow{w_1} C_2 \dots$

An input w is <u>accepted</u> by TM M if it reaches an accepting configuration from initial configuration q_{acc}

M <u>rejects</u> or does not accept w if
- M reaches q_{rej}
- M never halts

$L(M)$: set of strings accepted by M

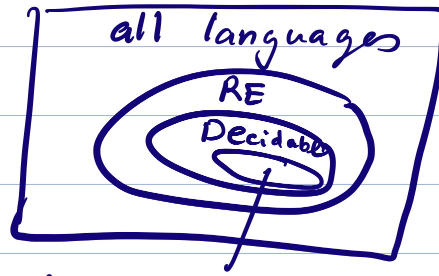
Church-Turing thesis.
Anything solvable using a mechanical procedure can be solved using a TM.

A Language L is <u>Recursively Enumerable</u> or Semi-decidable if there is a TM M such that $L = L(M)$.

A Language L is Recursive or <u>decidable</u> if there is a TM M that halts on all input and $L = L(M)$.

L is undecidable if it is not decidable.

Now we will use the source code of a TM M as the input to another TM.



Source code of M is denoted $\langle M \rangle$ NP, PSpace

There are language that are not RE

E.g. $L_d = \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$

<u>Proof</u> : Suppose L_d is RE. Then $\exists M_1$ such that $L(M_1) = L_d$

$$= \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$$

Question $\langle M_1 \rangle \in L(M_1)$?

Suppose $\langle M_1 \rangle \in L(M_1)$

by L_d $\langle M_1 \rangle \notin L(M_1)$

Suppose $\langle M_1 \rangle \notin L(M_1)$

by def L_d $\langle M_1 \rangle \in L(M_1)$

There is a language $L \in RE / REC$

$L_{halt} = \{ \langle M \rangle \mid M \text{ halts on } \varepsilon \}$

Halting problem is undecidable

How to show L_{RHA} is undecidable?

Encoding of a RHA $\mathcal{A}$ and mode $l \in L$
such that $\mathcal{A}$ reaches l

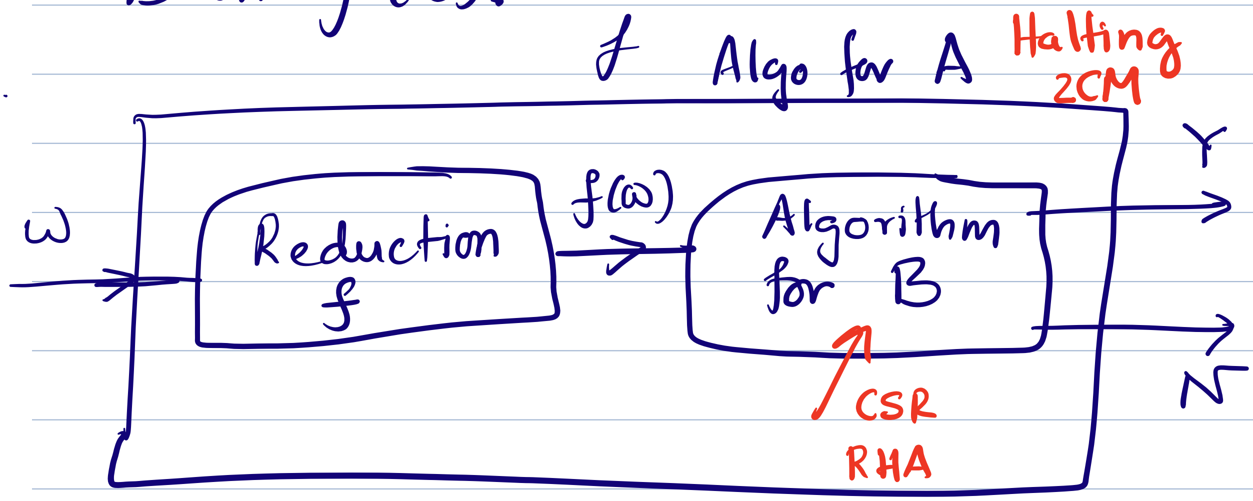
$L_{RHA} = \{ \langle \mathcal{A}, l \rangle \mid \mathcal{A} \text{ reaches } l \}$

To solve problem A we often use a subroutine for solving problem B.
E.g. Reachability of MRA solved by translating MRA to ITA + Reachability of ITA.

We can use the same idea to show hardness of problems.

Suppose a TM for deciding L_{RHA} could be used to solve L_{halt} then we could conclude that L_{RHA} is also undecidable. $\forall x \in L_{RHA} \quad f(x) \in L_{TM}$

Reductions : A reduction from A to B is a function f from instances of A to instances B such that solving A on x is same as solving B on $f(x)$.



Thus A is no harder than B.

$A \leq B$

Our Application

A : Halting problem for 2CM

B : CSR for RHA

$\boxed{\text{HALT} \leq \text{2-Stack-halt} \leq \cancel{\text{2-CM}}} \leq \text{RHA}$

2 Counter Machines (2CM)

- Finite program
- Two counters C, D initialized to 0

Instructions allowed

INCC, DECC
INCD, DECD
JNZC, JNZD

Example : 2CM program for 2x3

1. INCC
2. INCC // $C \leftarrow 2$
3. INCD
4. INCD
5. INCD
6. DECC
7. JNZC 3
8. // Halting location

Const

K counters → C

←

"Prime factorization"

Decision problem : Given a 2CM T
does T halt?

$L_{2CMHALT}$: Undecidable [Minsky 67]

2CM can simulate arbitrary TMs.

Reduction

f : 2CM-HALT $\longrightarrow$ RHA

/

Program counter PC	Locations / modes
Counters C, D	Clocks c, d
Instructions INCC, JNZD ..	Constant rate
Halting location	Transitions
	Halting location

Idea of reduction

Two clocks c, d will "Simulate" two counters C, D.

- $c = k_1 \left(\frac{k_2}{k_1}\right)^C \qquad d = k_1 \left(\frac{k_2}{k_1}\right)^D$

  $C = 0 \Leftrightarrow c = k_1$

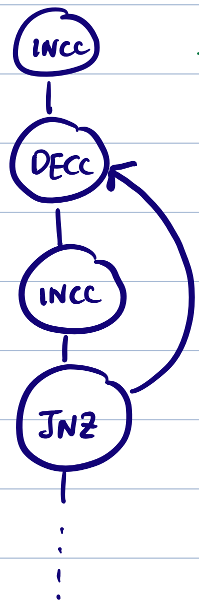
where $W > k_1 > k_2 > 0$ are constant rates for clocks

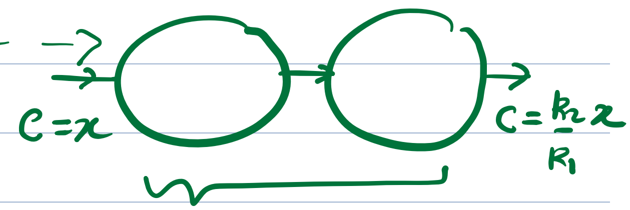
INCC : $C \leftarrow C+1 \qquad c = k_1 \left(\frac{k_2}{k_1}\right)^{C+1} = c\,\frac{k_2}{k_1}$

DECC : $C \leftarrow C-1 \qquad c = k_1 \left(\frac{k_2}{k_1}\right)^{C-1} = c\,\frac{k_1}{k_2}$

after checking $c < k_1$

2CM $\longrightarrow f \longrightarrow$ RHA

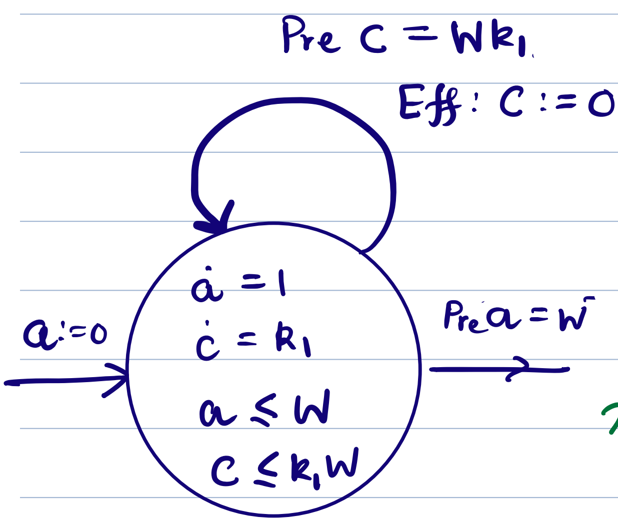


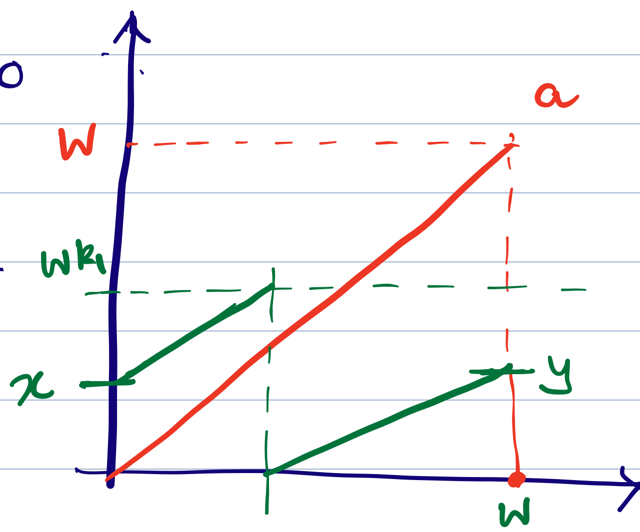


automaton fragment or widget

We have to provide widgets for INCC, DECC, INCD, JNZC, ... and then we are done.

A building block Widget that preserves the value of clock c.

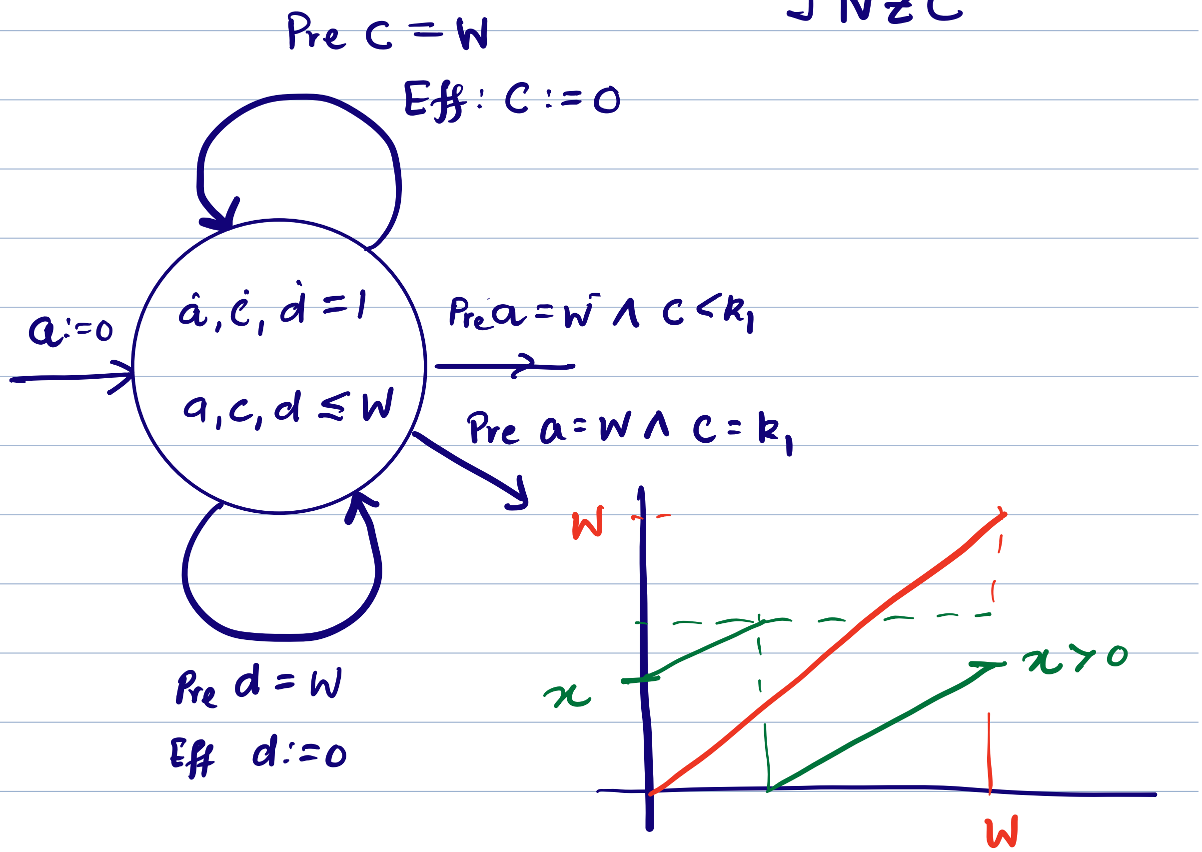




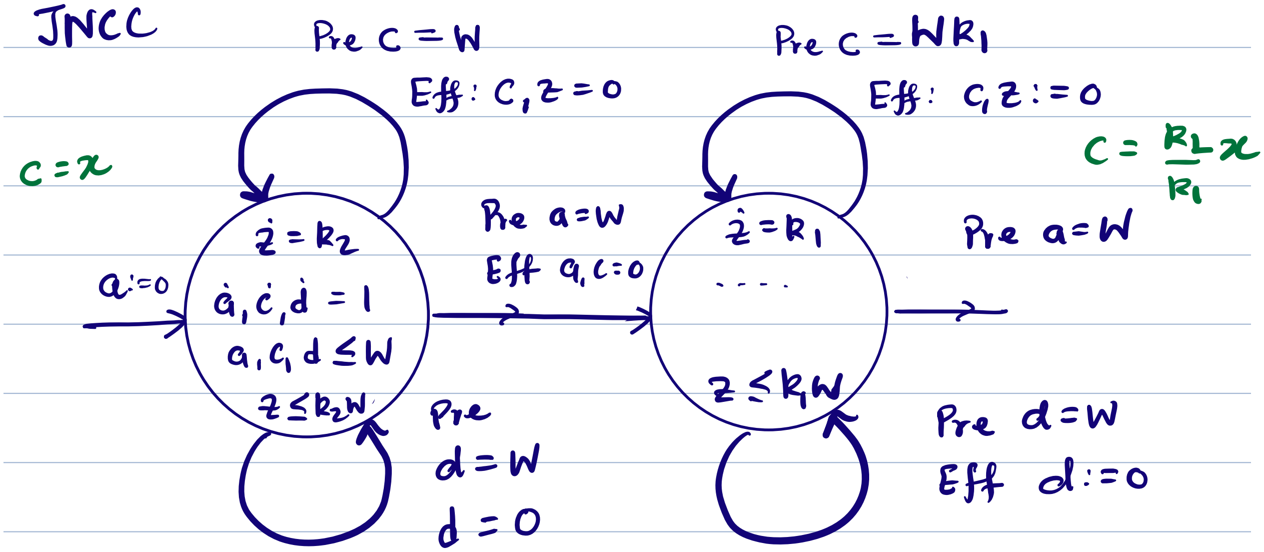
$$t = \frac{Wk_1 - x}{k_1} = W - \frac{x}{k_1}$$

$$y = (W - t) \cdot k_1 = \left(W - W + \frac{x}{k_1}\right) k_1$$
$$= x$$

JNZC



JNCC



Reduction from 2CM to RHA.

Putting it all together

Control state Reachability of RHA is undecidable.