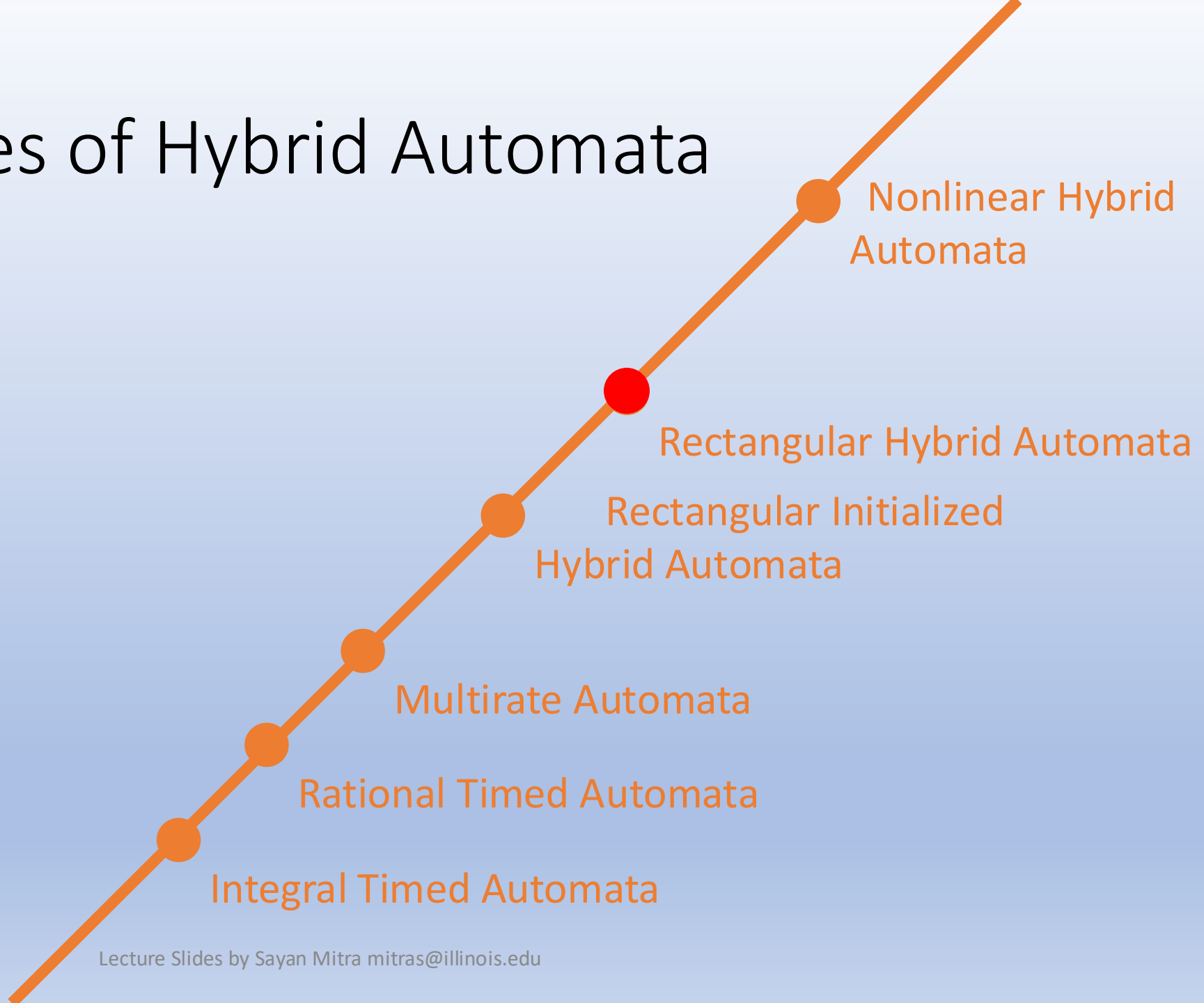
Undecidability of Rectangular Hybrid Automata

Sayan Mitra

Verifying cyberphysical systems

mitras@illinois.edu

Special Classes of Hybrid Automata



Control State Reachability of Rectangular HA

Definition. A **rectangular hybrid automaton (RHA)** is a HA $\mathcal{A} = \langle V, Q_0, \mathcal{T}, \mathcal{D} \rangle$ where

- V = X ∪ $\{loc\}$, X: **continuous variables**, loc discrete variable of finite type Ł
- $Q_0 \subseteq Val(V)$ set of initial states
- $\mathcal{T} = \cup_\ell \mathcal{T}_\ell$ set of trajectories for X
  - For each $\tau \in \mathcal{T}_\ell, x \in X$ either (i) $d(x) = k_\ell$ or (ii) $d(x) \in [k_{\ell 1}, k_{\ell 2}]$
- $\mathcal{D}$ is a set of transitions such that
  - Guards are described by **rational** clock constraings
  - $\langle x, l \rangle \rightarrow_a \langle x', l' \rangle$ implies $x' = x \; or x' \in [c_1, c_2]$

Control state reachability problem: Is $loc = \ell^*$ reachable from Q_0?

Theorem. Control state reachability problem for RHA is undecidable.

Henzinger, Kopke, Puri, and Varaiya. What's Decidable About Hybrid Automata?. Journal of Computer and System Sciences. ACM Press, 1995.

Implications

- There exists no algorithm that terminates and answers this fundamental verification question of rectangular hybrid automata
- **Use less expressive models**. CSR problem for initialized rectangular hybrid automata (IRHA) is decidable. Express or approximate your problem as IRHA
- **Use approximation algorithms for partial decisions.** Reachability analysis algorithms can over-approximate $\mathrm{Reach}(\mathcal{A}, Q_0) \subseteq \boldsymbol{R}$. If $R \cap Unsafe = \emptyset$ then we can conclude that RHA $\mathcal{A}$ is safe, but otherwise we cannot conclude that $\mathcal{A}$ is unsafe.

Outline

- Hardness results
- Halting problem
- Reductions
- Counter machines
- RHA

One problem that is hard / undecidable

Halting Problem is Undecidable: Diagonalization

- **Theorem**. There does not there exist an algorithm/compiler HALTS(P, x) that correctly answers yes iff a program P halts on x.
- Assume, for contradiction, there is a correct procedure:
- HALTS(P, x) returns:
  - TRUE if P halts on input x
  - FALSE if P runs forever on input x
- Now define a new program D(P):

  if HALTS(P, P) = TRUE then loop forever
  else halt immediately
- What happens when we run D on itself, D(D)?
  - If HALTS(D, D) = TRUE, then D(D) should halt
    but by definition of D, in this case D(D) loops forever. Contradiction.
  - If HALTS(D, D) = FALSE, then D(D) should loop forever…
    but by definition of D, in this case D(D) halts. Contradiction.

Cantor: build a real number that differs from the n-th real at the n-th digit.

Halting proof: build a program that differs from the n-th program on input n.

How to show that RHA is undecidable?

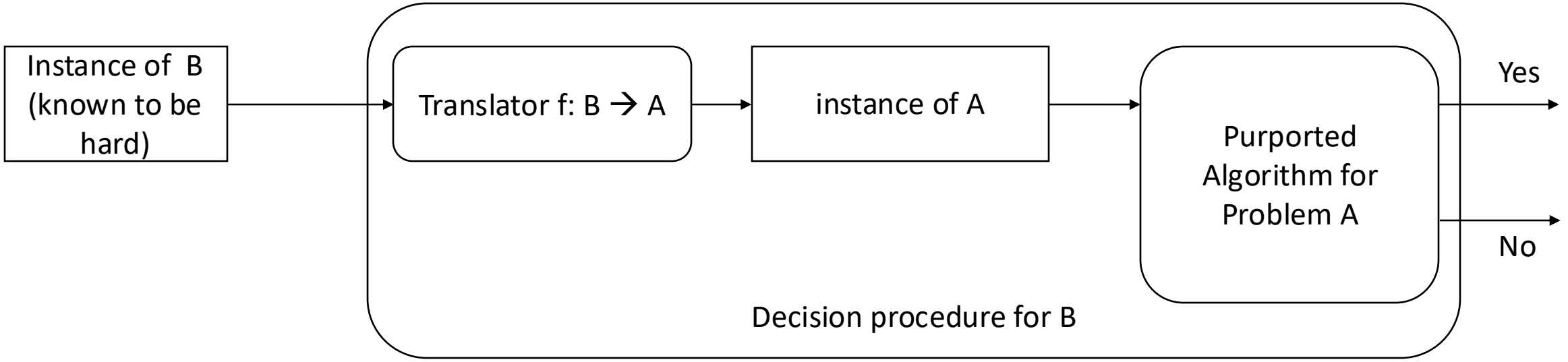
Show that if RHA were decidable then Halting problem also becomes decidable.

- Since Halting problem is undecidable, therefore CSR of RHA is also undecidable.

Do this in multiple steps

- If Halting problem for 2-Counter Machines were decidable then Halting problem would be decidable
  - 2 Counter Machines are as powerful/expressive as general programs
- If RHA were decidable then 2-Counter machines would be decidable
  - RHA are as expressive / powerful as 2-CMs

General reductions: Using known hard problem B to show hardness of A

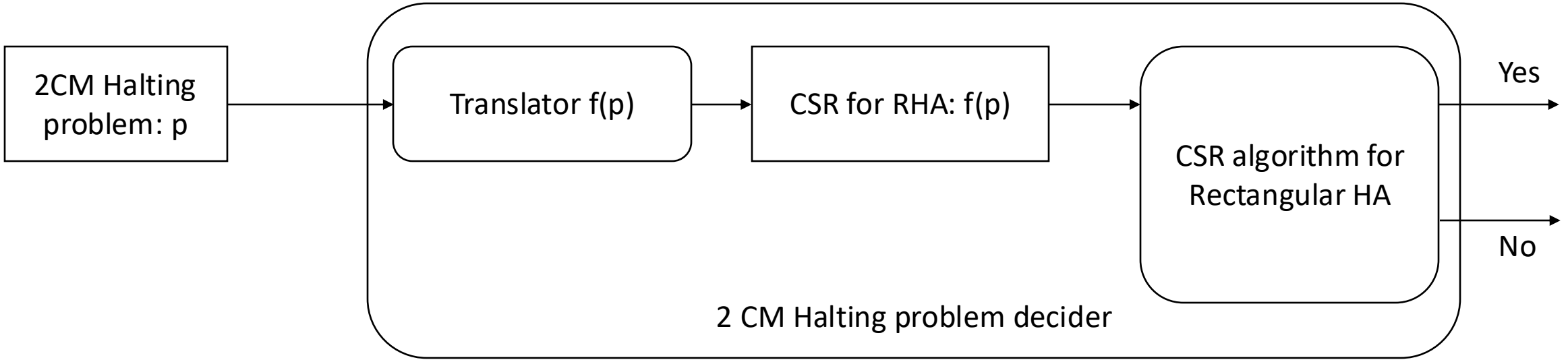


Given B is known to be hard

Suppose (for the sake of contradiction) A is solvable

If we can construct a reduction f: B→A (from B to A) then B becomes easy, which is a contradiction

Reduction from Halting Problem for 2CM



Suppose CSR for RHA is decidable

If we can construct a reduction from 2CM Halting Problem to CSR for RHA then 2CM Halting problem is also decidable

Counter Machines

An n-counter machine is an elementary computer with n-unbounded counters and a finite program written in a minimalistic assembly language.

More precisely: A 2-counter machine (2CM) is a discrete transition system with the following components:

- Two nonnegative integer counters C and D. Both are initialized to 0.
- A finite program with one of these instructions at each location (or line):
  - INCC, INCD: increments counter C (or D)
  - DECC, DECD: decrements counter C (or D), provided it is not 0,
  - JNZC, JNZD [label]: moves the program control to line *label* provided that counter C (or D) is not zero.

Example 2CM for multiplication

A 2-counter machine for multiplying 2x3 is shown below.

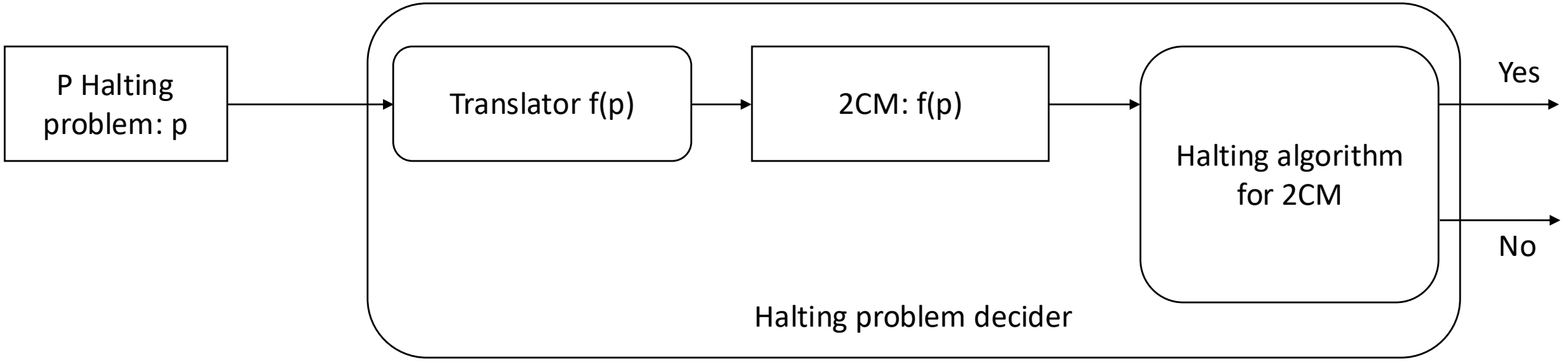
```
INCC;
INCC;          % C = 2
INCD;          % LOOP
INCD;
INCD;
DECC;
JNZC 3;        % Jump to LOOP
               % HALT
```

Exercise: Show that any k-counter machine can be simulated by a 2CM.

Halting problem for 2CM

- A **configuration** of a 2CM is a triple (pc, C, D)
  - pc is the program counter that stores the next line to be executed
  - C, D are values of the counter
- A sequence of configurations (pc0, D0, C0), (pc1, D1, C1), … is an **execution** if the ith configuration goes to the (i+1)st configuration in the sequence executing the instruction in line pci
- Given a 2CM **M** a special halting location (pc_halt), the Halting problem requires us to decide whether all executions of **M** reach the halting location
- Theorem [Minsky 67]. The Halting problem for 2CMs is undecidable.

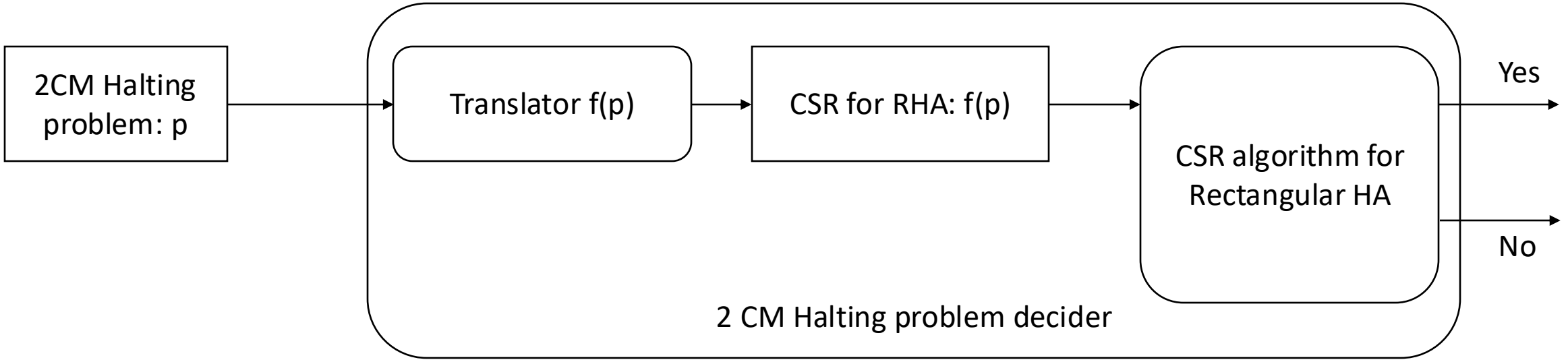
Reduction from Halting Problem



Reduction from 2CM to CSR-RHS

We have to construct a function (reduction) that maps instances of 2CM-Halt to instances of CSR-RHA

Reduction from Halting Problem for 2CM



Suppose CSR for RHA is decidable

If we can construct a reduction from 2CM Halting Problem to CSR for RHA then 2CM Halting problem is also decidable

Translation/Reduction from 2CM to CSR-RHS

- Program counter pc
- Counters C, D
- Instructions (program)
- Halting location

- Locations, sequence of locations
- Clocks c, d that can go at some constant rates $k_1, k_2, \ldots$
- Transitions: *widgets*
- Particular location / control state (to which we will check CSR)

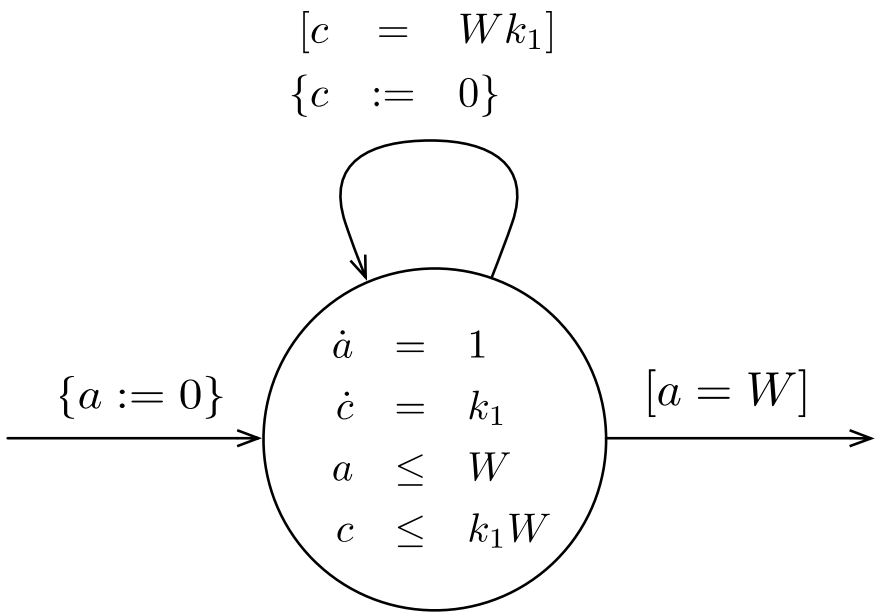
Idea of reduction: a compiler from 2CM->RHA

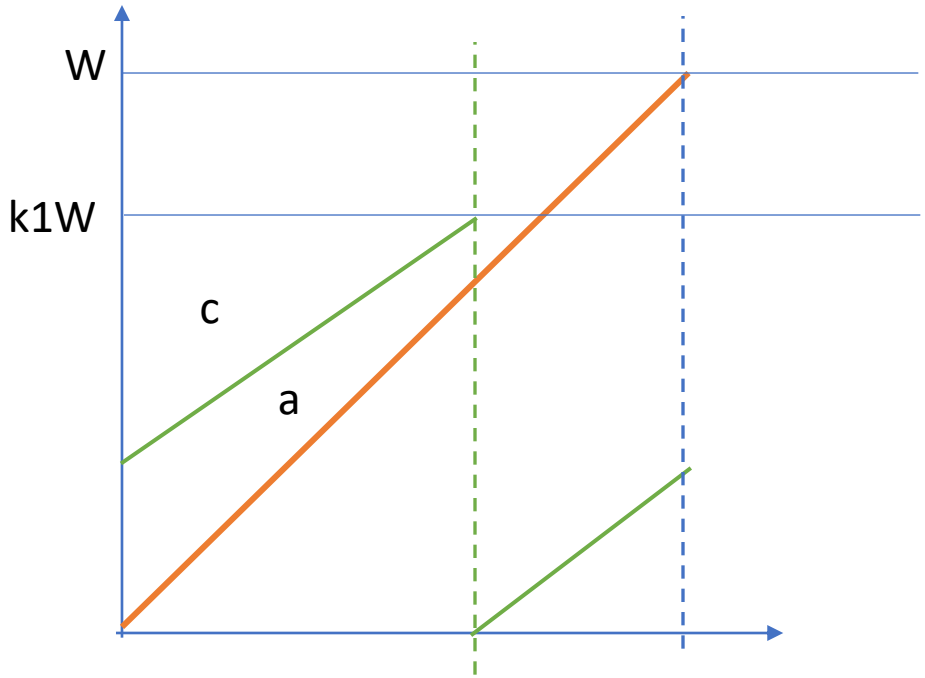
- Two clocks
- $c = k_1 \left(\frac{k_2}{k_1}\right)^C$
- $\mathrm{d} = k_1 \left(\frac{k_2}{k_1}\right)^D$

- INCC
- $k_1 \left(\frac{k_2}{k_1}\right)^{C+1} = c \left(\frac{k_2}{k_1}\right)$

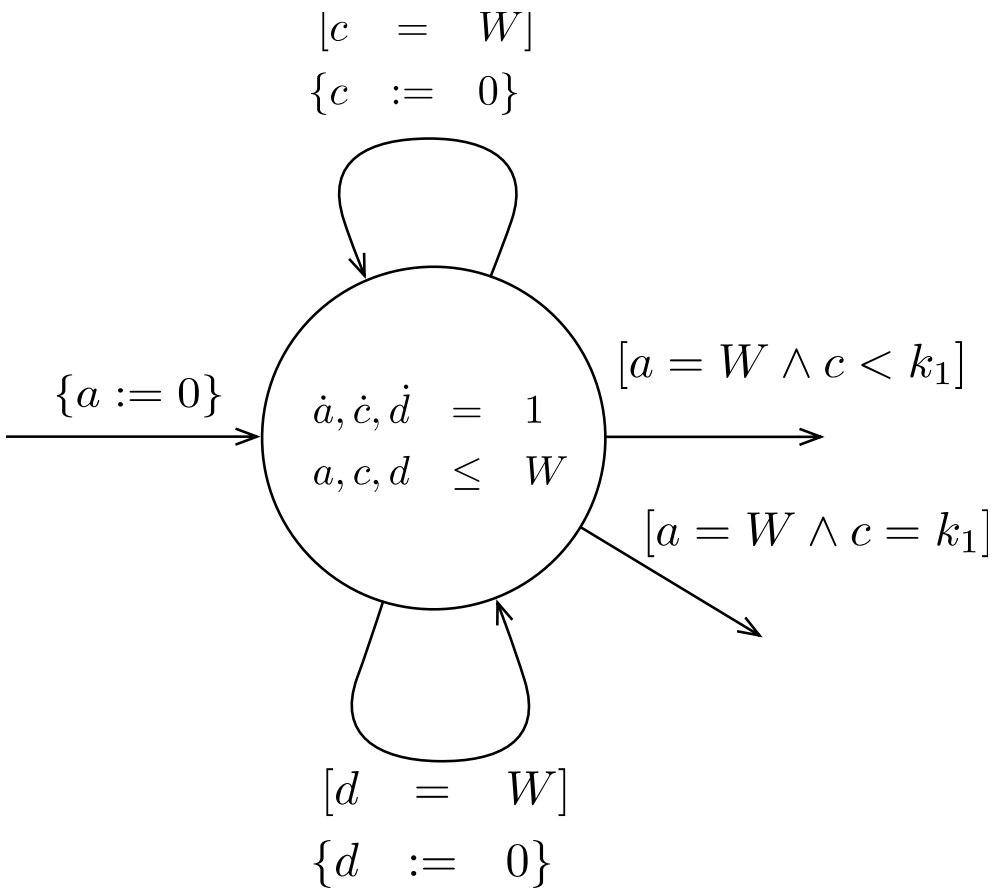
- DECC
- $k_1 \left(\frac{k_2}{k_1}\right)^{C-1} = c \left(\frac{k_1}{k_2}\right)$ after checking nonzero $c < k_1$

A widget that preserves the value of clock c

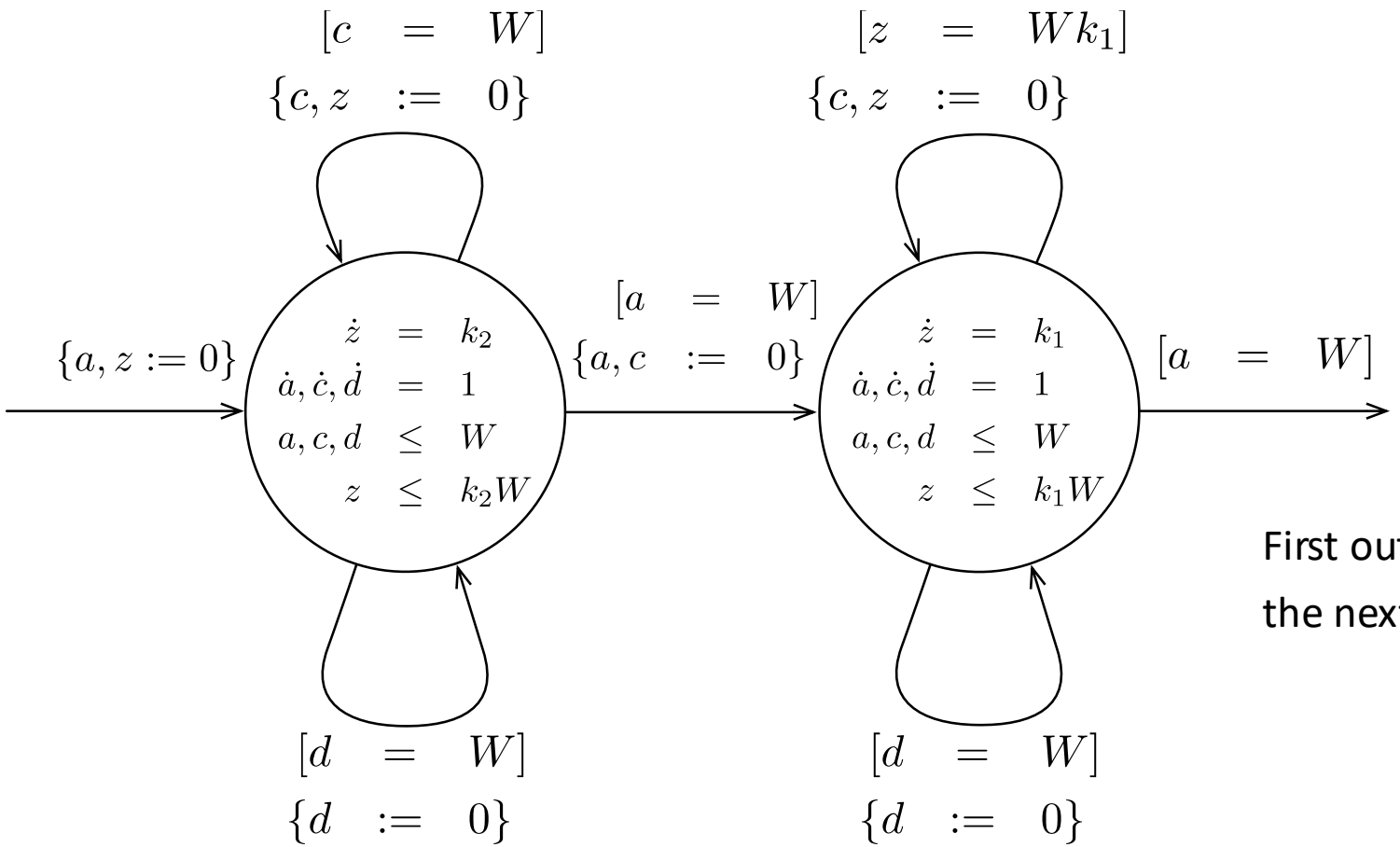




A widget for checking JNZC (c < k_1)

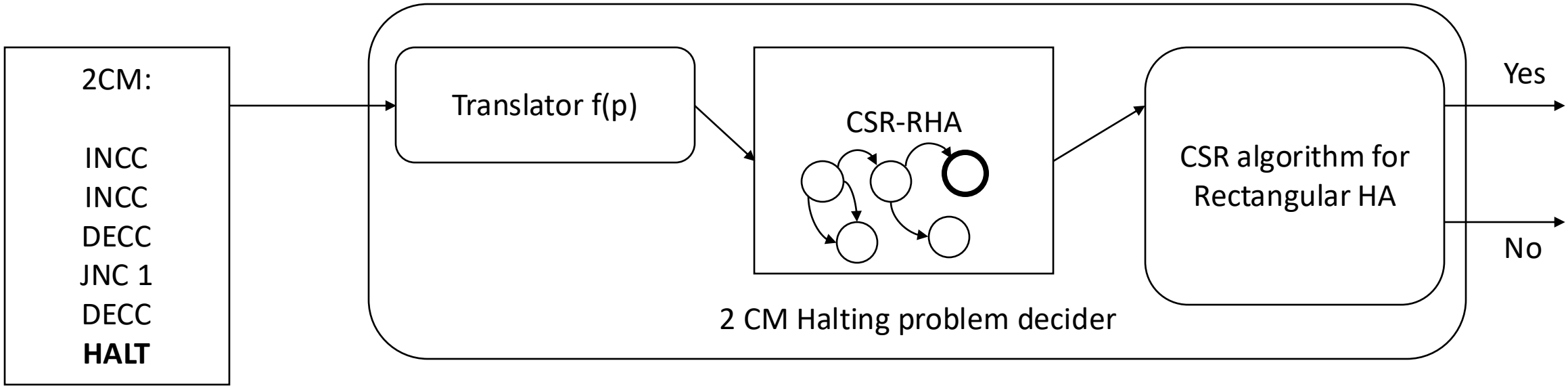


A widget implementing INCC



First outgoing transition sets $z = k_2 c$ and the next outgoing transition sets c $= z * \left(\frac{1}{k_1}\right)$

Putting it all together



Suppose CSR for RHA is decidable

If we can construct a reduction from 2CM Halting Problem to CSR for RHA then 2CM Halting problem is also decidable

Theorem: CSR for RHA is undecidable