Playing with repeating values in datawords using energy games

Diego Figueira and M. Praveen

Outline

Outline

Specifications for a coffee machine

- Whenever coffee button is pressed, coffee is produced in the next step.
- Whenever stop button is pressed, no coffee is produced in the next step.

Specifications for a coffee machine

- Whenever coffee button is pressed, coffee is produced in the next step.
- Whenever stop button is pressed, no coffee is produced in the next step.
- Specifications satisfiable:

$$\begin{array}{rl} \text{coffee button} & \bot \\ \text{stop button} & \bot \\ \text{make coffee} & \bot \end{array}$$

Specifications for a coffee machine

- Whenever coffee button is pressed, coffee is produced in the next step.
- Whenever stop button is pressed, no coffee is produced in the next step.
- Specifications satisfiable:

coffee button	$\bot$	$\bot$	$\bot$	$\cdots$
stop button	$\bot$	$\bot$	$\bot$	$\cdots$
make coffee	$\bot$	$\bot$	$\bot$	$\cdots$

Realizability of specifications

Coffee button and stop button are not under the control of the system.

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button
stop button

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button
stop button
make coffee

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$
stop button	$*$
make coffee	

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$
stop button	$*$
make coffee	$*$

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$
stop button	$*$	$*$
make coffee	$*$	

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$
stop button	$*$	$*$
make coffee	$*$	$*$

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$
stop button	$*$	$*$	$*$
make coffee	$*$	$*$	

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$
stop button	$*$	$*$	$*$
make coffee	$*$	$*$	$*$

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$	$\cdots$
stop button	$*$	$*$	$*$	$\cdots$
make coffee	$*$	$*$	$*$	

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$	$\cdots$
stop button	$*$	$*$	$*$	$\cdots$
make coffee	$*$	$*$	$*$	$\cdots$

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$	$\cdots$
stop button	$*$	$*$	$*$	$\cdots$
make coffee	$*$	$*$	$*$	$\cdots$

The realizability problem:

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$	$\cdots$
stop button	$*$	$*$	$*$	$\cdots$
make coffee	$*$	$*$	$*$	$\cdots$

The realizability problem:
Input: A formula, a partition of the variables

Realizability of specifications

Coffee button and stop button are not under the control of the system.

coffee button	$*$	$*$	$*$	$\cdots$
stop button	$*$	$*$	$*$	$\cdots$
make coffee	$*$	$*$	$*$	$\cdots$

The realizability problem:
Input: A formula, a partition of the variables
Question: Does the system have a winning strategy?

Outline

Model, syntax, semantics

$$
\begin{array}{lccccccc}
x: & d_1 & \$ & * & \cdots & * & * & \cdots \\
y: & d_2 & * & \$ & \cdots & \# & \$ & \cdots \\
w: & \top & \bot & \top & \cdots & * & * & \cdots
\end{array}
$$

Model, syntax, semantics

$$
\begin{array}{lccccccc}
x: & d_1 & \$ & * & \cdots & * & * & \cdots \\
y: & d_2 & * & \$ & \cdots & \# & \$ & \cdots \\
w: & \top & \bot & \top & \cdots & * & * & \cdots
\end{array}
$$

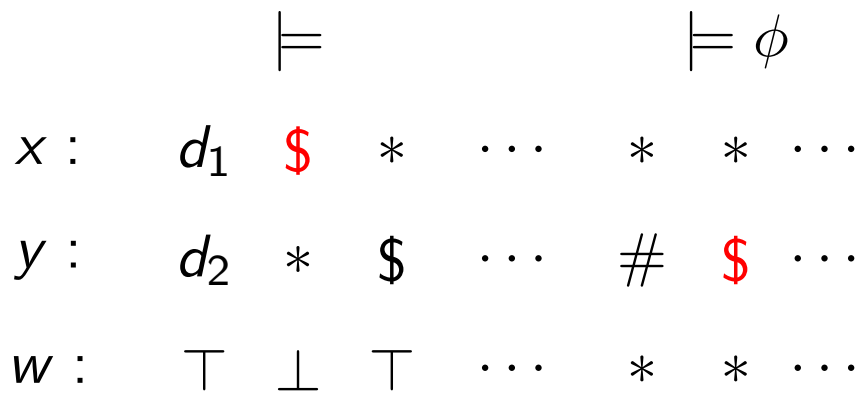
$$
\begin{aligned}
\phi ::= & x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid \\
& w \mid \neg \phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi
\end{aligned}
$$

Model, syntax, semantics

$$
\begin{array}{lcccccccc}
 & & \models & & & & & & \\
x: & d_1 & \$ & * & \cdots & * & * & \cdots \\
y: & d_2 & * & \$ & \cdots & \# & \$ & \cdots \\
w: & \top & \bot & \top & \cdots & * & * & \cdots
\end{array}
$$

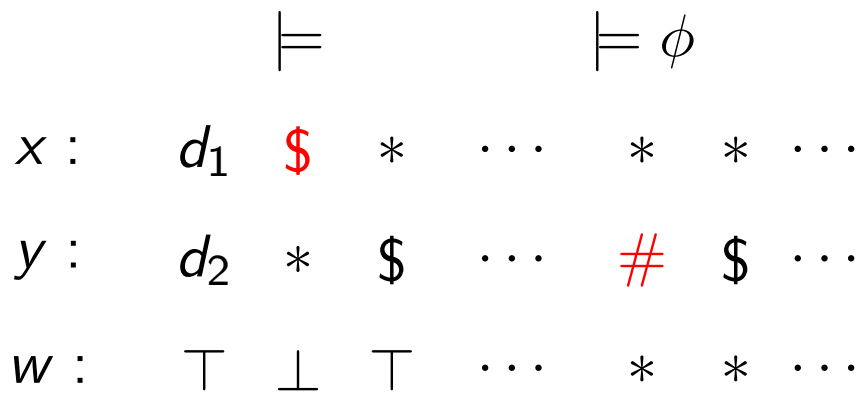
$$
\begin{aligned}
\phi ::= & x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid \\
& w \mid \neg \phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi
\end{aligned}
$$

Model, syntax, semantics



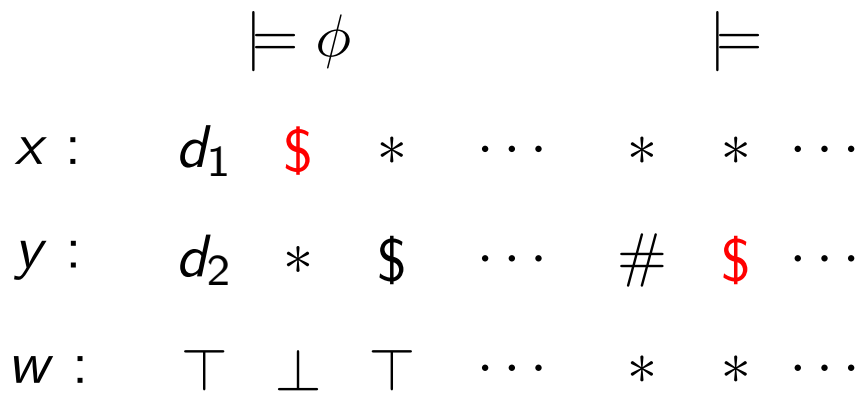
$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi? \rangle y \mid x \not\approx \langle \phi? \rangle y \mid y \approx \langle \phi? \rangle^{-1} x \mid y \not\approx \langle \phi? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

Model, syntax, semantics



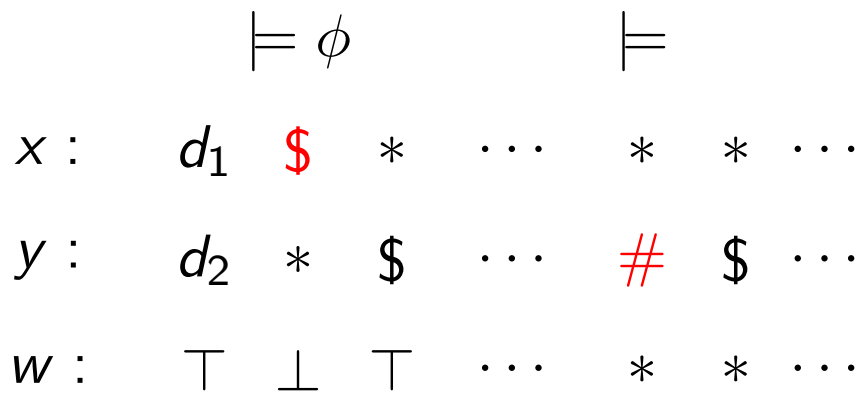
$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

Model, syntax, semantics



$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

Model, syntax, semantics



$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

Realizability of LRV formulas

- Realizability of propositional LTL: parity games on finite graphs.

Realizability of LRV formulas

- Realizability of propositional LTL: parity games on finite graphs.

- Satisfiability of LRV: reachability in VASS [Demri, D'Souza, Gascon 2007].

Realizability of LRV formulas

- Realizability of propositional LTL: parity games on finite graphs.
- Satisfiability of LRV: reachability in VASS [Demri, D'Souza, Gascon 2007].
- Realizability of LRV: parity games on VASS.

Outline

Asymmetry in games on VASS

- [Raskin, Samuelides, Van Begin 2005] One of the palyers has transitions that are downward closed. Coverability games decidable.

Asymmetry in games on VASS

- [Raskin, Samuelides, Van Begin 2005] One of the palyers has transitions that are downward closed. Coverability games decidable.

- [Abdulla, Bouajjani, D'orso 2008] One of the players has lossy transitions. Safety games are decidable.

Asymmetry in games on VASS

- [Raskin, Samuelides, Van Begin 2005] One of the palyers has transitions that are downward closed. Coverability games decidable.
- [Abdulla, Bouajjani, D'orso 2008] One of the players has lossy transitions. Safety games are decidable.
- [Brázdil, Jančar, Kučera 2010] Transitions can add arbitrarily large numbers. Decidable to check if one of the players can make some counter zero.

Asymmetry in games on VASS

- [Raskin, Samuelides, Van Begin 2005] One of the palyers has transitions that are downward closed. Coverability games decidable.
- [Abdulla, Bouajjani, D'orso 2008] One of the players has lossy transitions. Safety games are decidable.
- [Brázdil, Jančar, Kučera 2010] Transitions can add arbitrarily large numbers. Decidable to check if one of the players can make some counter zero.
- [Bérard, Haddad, Sassolas, Sznajder 2012] One palyer can only increment; the other player cannot test for zero.

Asymmetry in games on VASS

- [Chatterjee, Randour, Raskin 2013] Energy games: if a player makes a counter to go below zero, the other player wins immediately. One of the players has to additionally satisfy a parity condition.

Asymmetry in games on VASS

- [Chatterjee, Randour, Raskin 2013] Energy games: if a player makes a counter to go below zero, the other player wins immediately. One of the players has to additionally satisfy a parity condition.

- [Abdulla, Mayr, Sangnier, Sproston 2013] Single-sided VASS games: transitions that make some counter to go below zero are disabled for both players. One of the players cannot change counters; the other player has to additionally satisfy a parity condition.

Single-sided LRV games

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.

Single-sided LRV games

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.

Single-sided LRV games

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

Single-sided LRV games

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.
- Realizability can be reduced to single-sided VASS games.

Single-sided LRV games — symbolic models

Concrete model

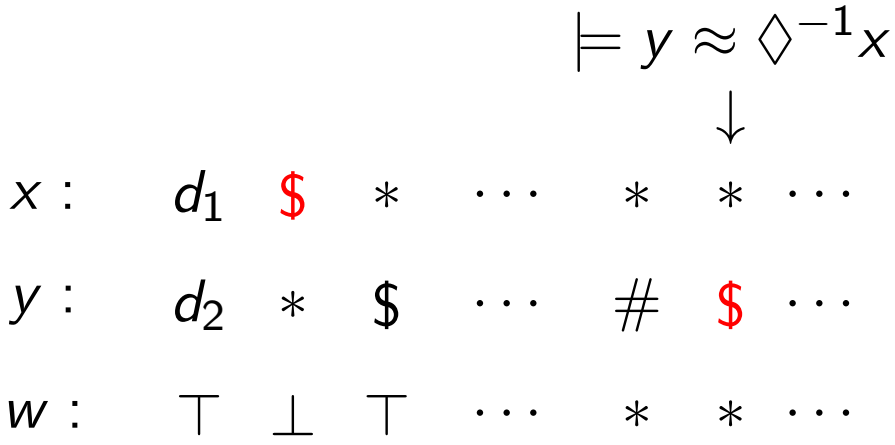
$$\models y \approx \Diamond^{-1} x$$

$$\downarrow$$

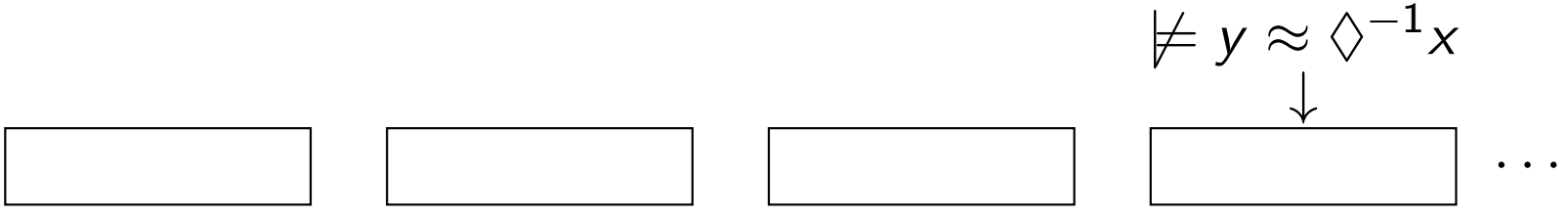
$$\begin{array}{lccccccc} x: & d_1 & \$ & * & \cdots & * & * & \cdots \\ y: & d_2 & * & \$ & \cdots & \# & \$ & \cdots \\ w: & \top & \bot & \top & \cdots & * & * & \cdots \end{array}$$

Single-sided LRV games — symbolic models

Concrete model

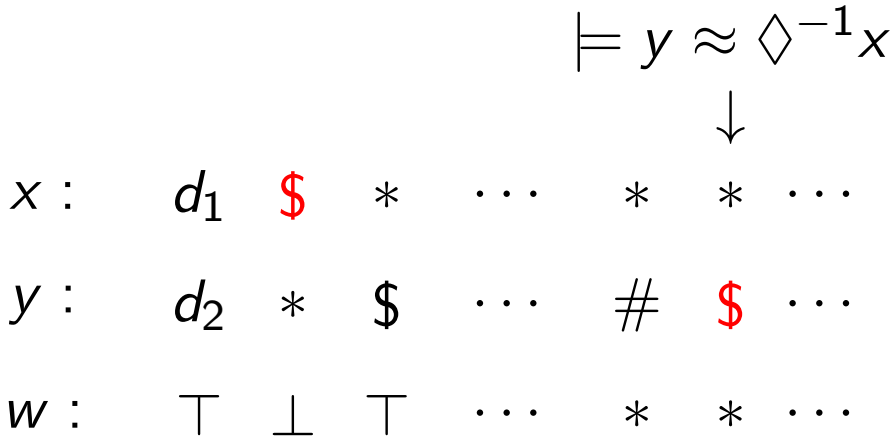


Symbolic model

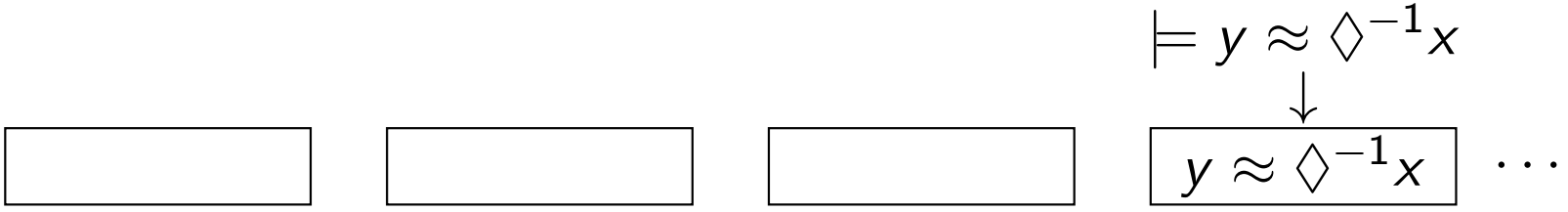


Single-sided LRV games — symbolic models

Concrete model



Symbolic model



Symbolic models

$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg \phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

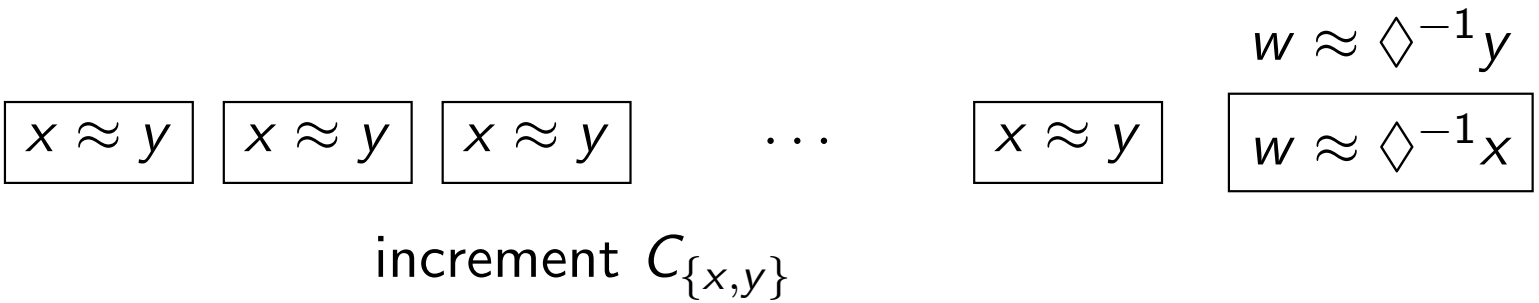
Symbolic models

$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$

$$\boxed{x \approx y} \quad \boxed{x \approx y} \quad \boxed{x \approx y} \quad \cdots \quad \boxed{x \approx y} \quad \begin{matrix} w \approx \Diamond^{-1} y \\ \boxed{w \approx \Diamond^{-1} x} \end{matrix}$$

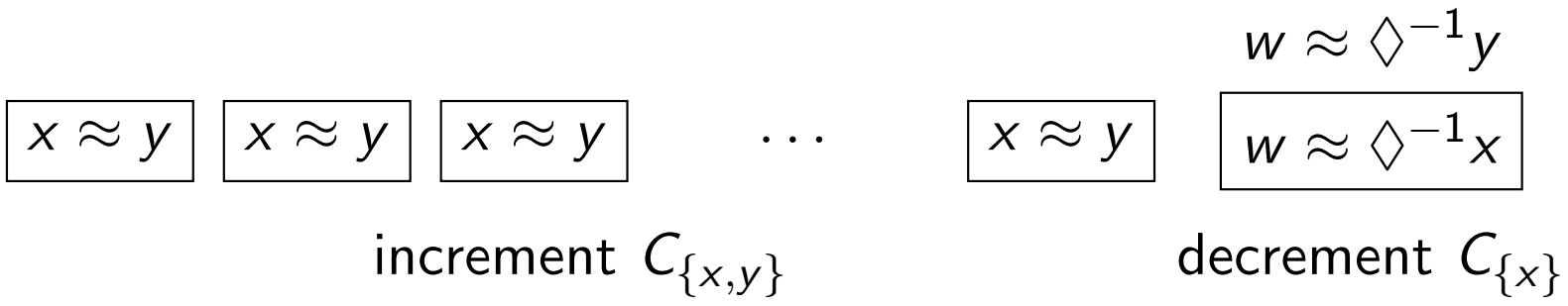
Symbolic models

$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$



Symbolic models

$$\phi ::= x \approx \mathsf{X}^1 y \mid x \approx \langle \phi ? \rangle y \mid x \not\approx \langle \phi ? \rangle y \mid y \approx \langle \phi ? \rangle^{-1} x \mid y \not\approx \langle \phi ? \rangle^{-1} x \mid$$
$$w \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$$



Outline

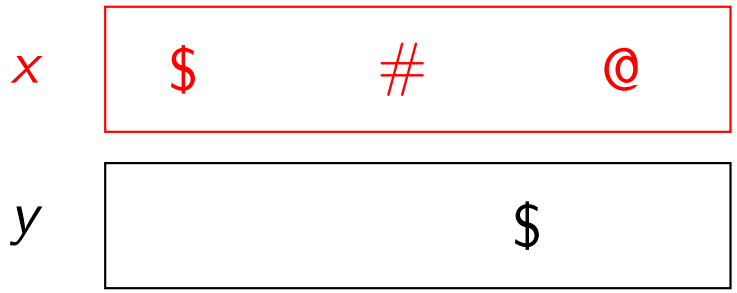
Restrictions to get decidability

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

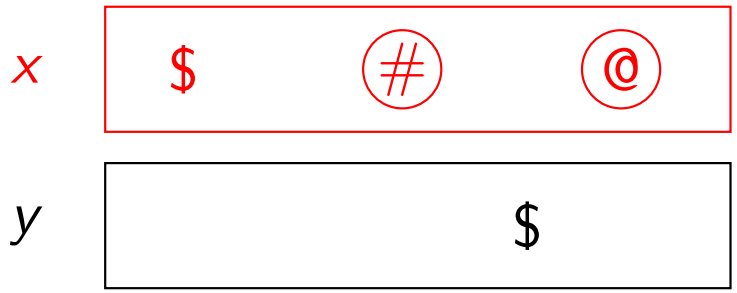
Restrictions to get decidability

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- ~~Environment player has only Boolean variables.~~
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

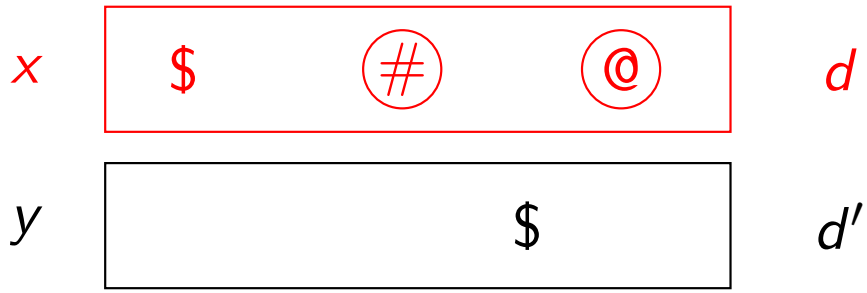
Simulating counter machines



Simulating counter machines

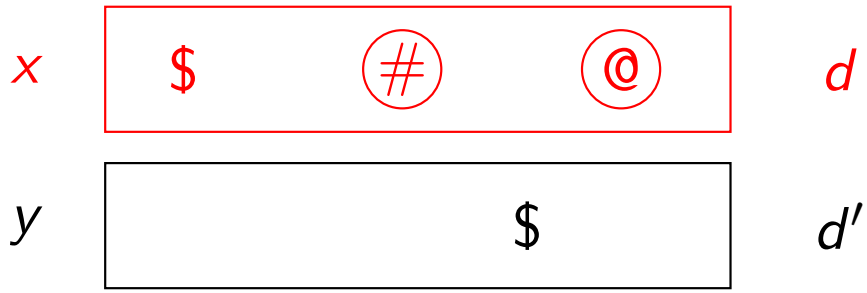


Simulating counter machines



Increment

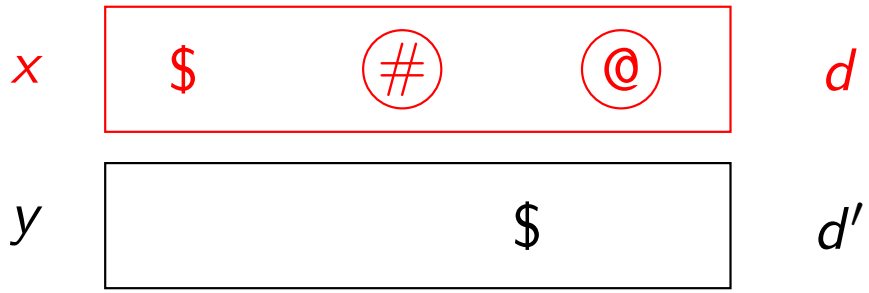
Simulating counter machines



Increment

- d doesn't repeat in the past in x or y

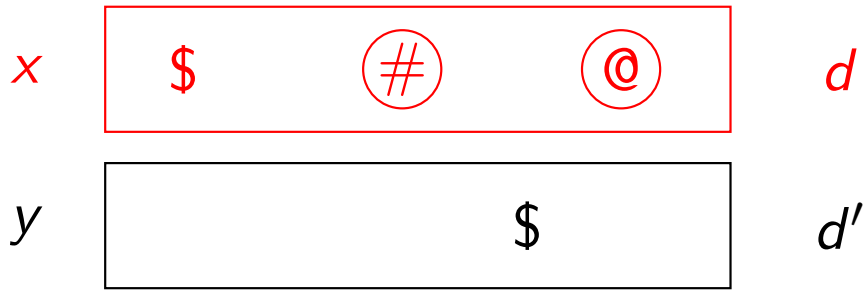
Simulating counter machines



Increment

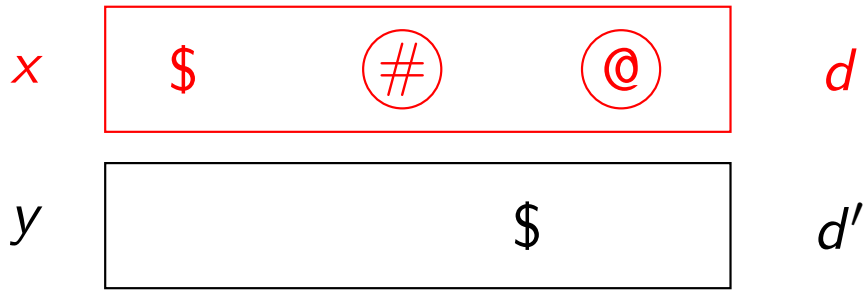
- d doesn't repeat in the past in x or y
- d' appears in the past in x and y

Simulating counter machines



Decrement

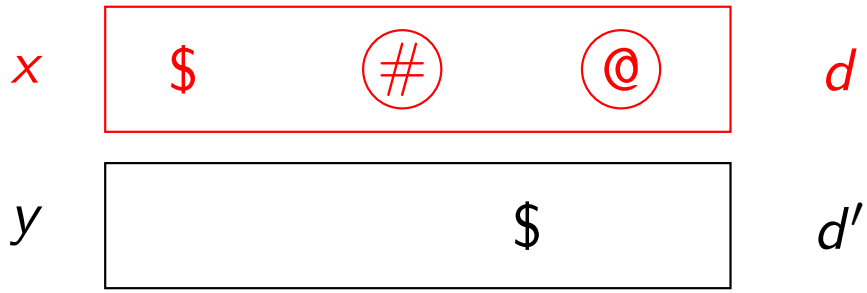
Simulating counter machines



Decrement

- d should appear in the past in x but not in y

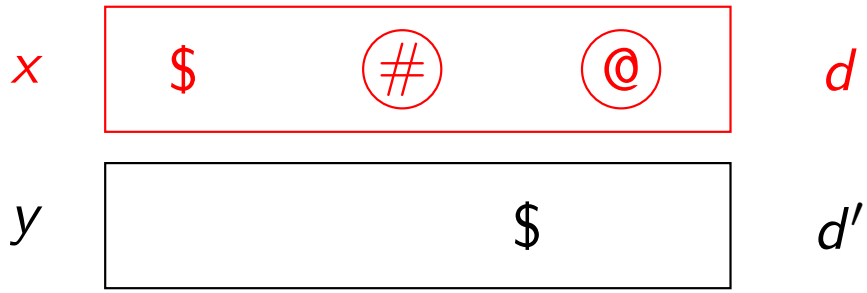
Simulating counter machines



Decrement

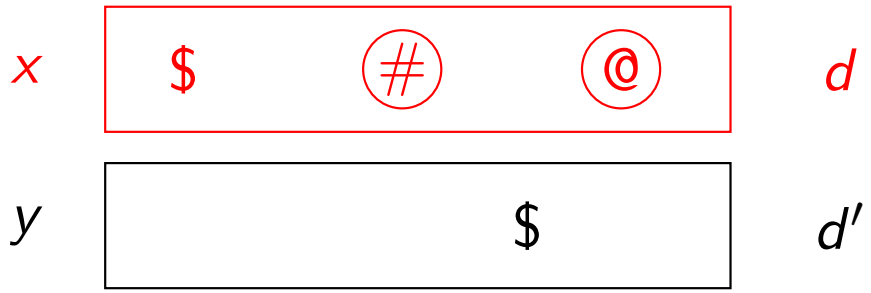
- d should appear in the past in x but not in y
- $d' = d$

Simulating counter machines



Zero test

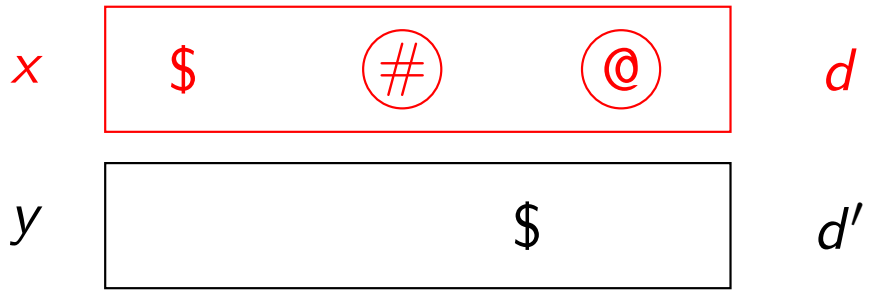
Simulating counter machines



Zero test

- In case of no cheating, $d = d'$ and no repetition in the past.

Simulating counter machines



Zero test

- In case of no cheating, $d = d'$ and no repetition in the past.
- In case of cheating, second player can win immediately by ensuring d' repeats in the past in x but not in y.

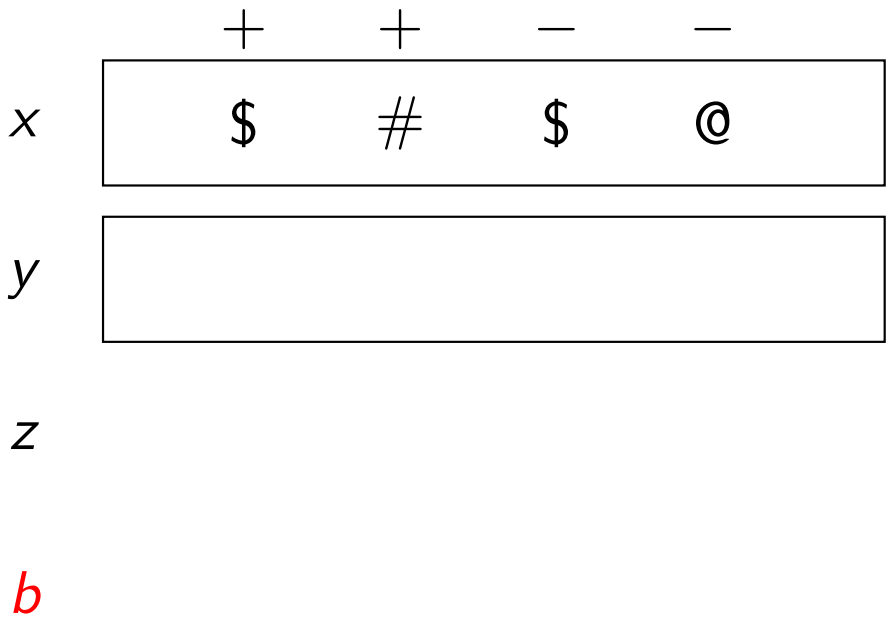
Restrictions to get decidability

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

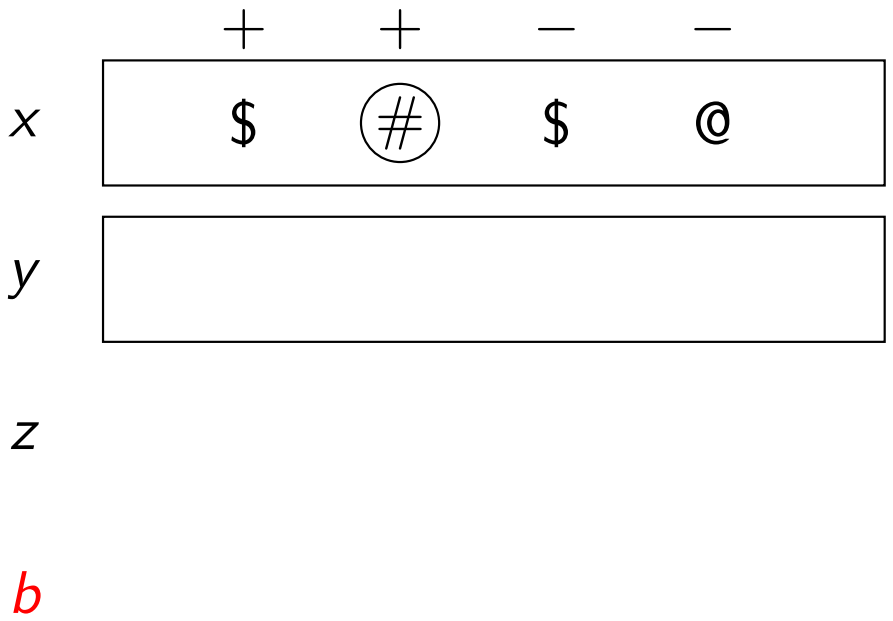
Restrictions to get decidability

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- ~~No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.~~

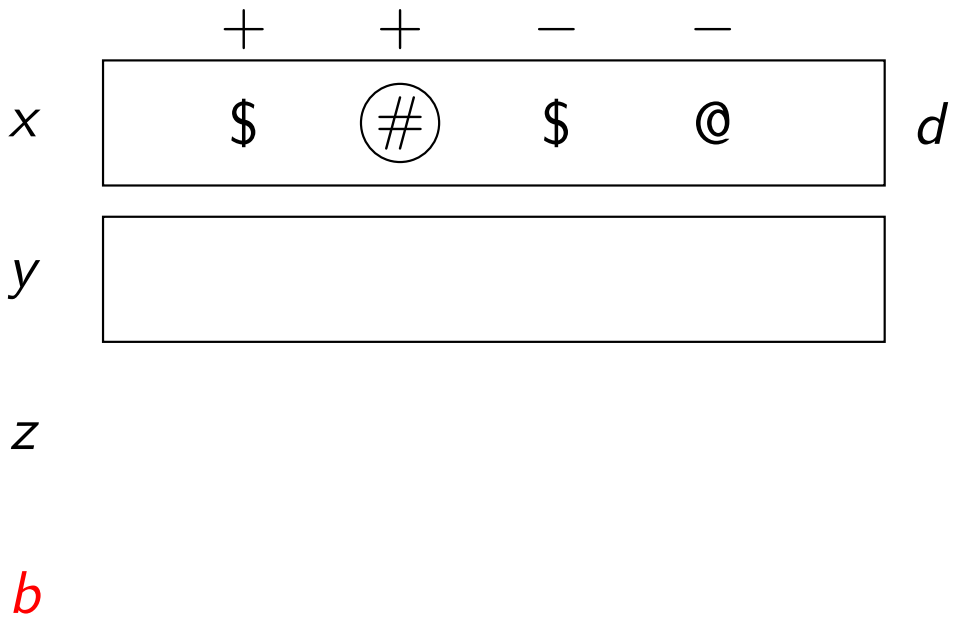
Simulating counter machines



Simulating counter machines

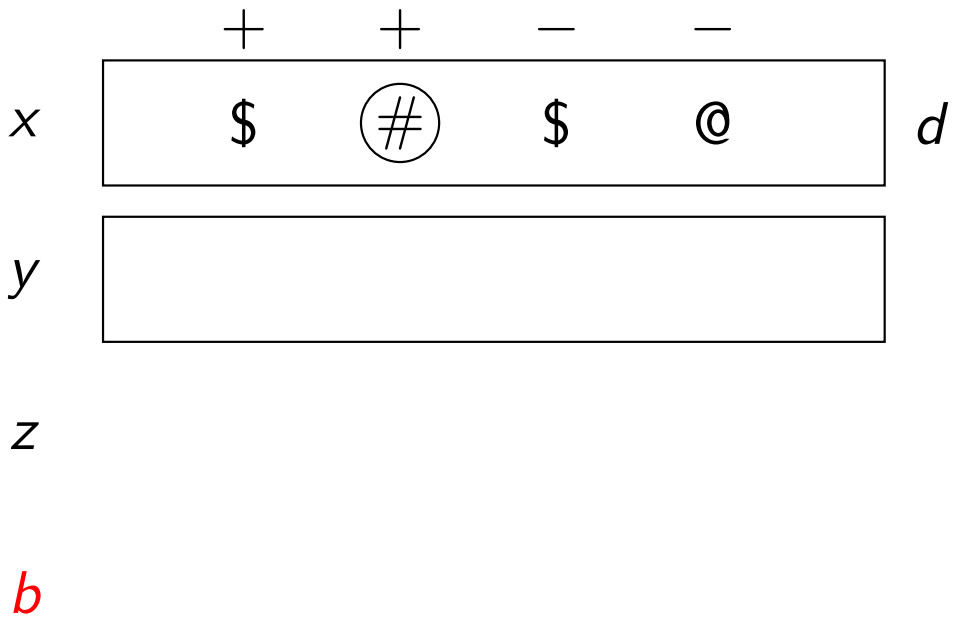


Simulating counter machines



Increment

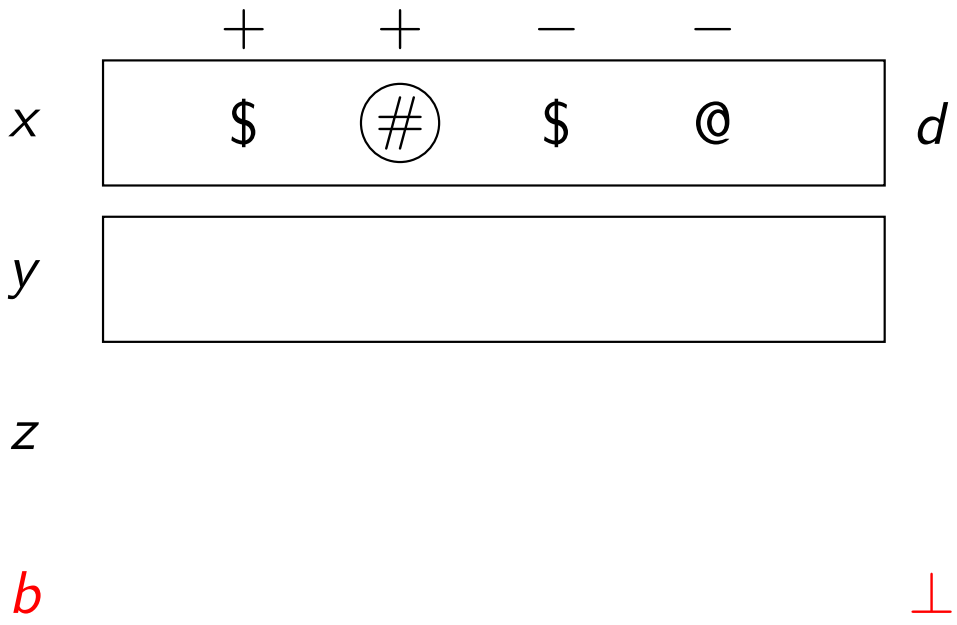
Simulating counter machines



Increment

- d should be a new data value.

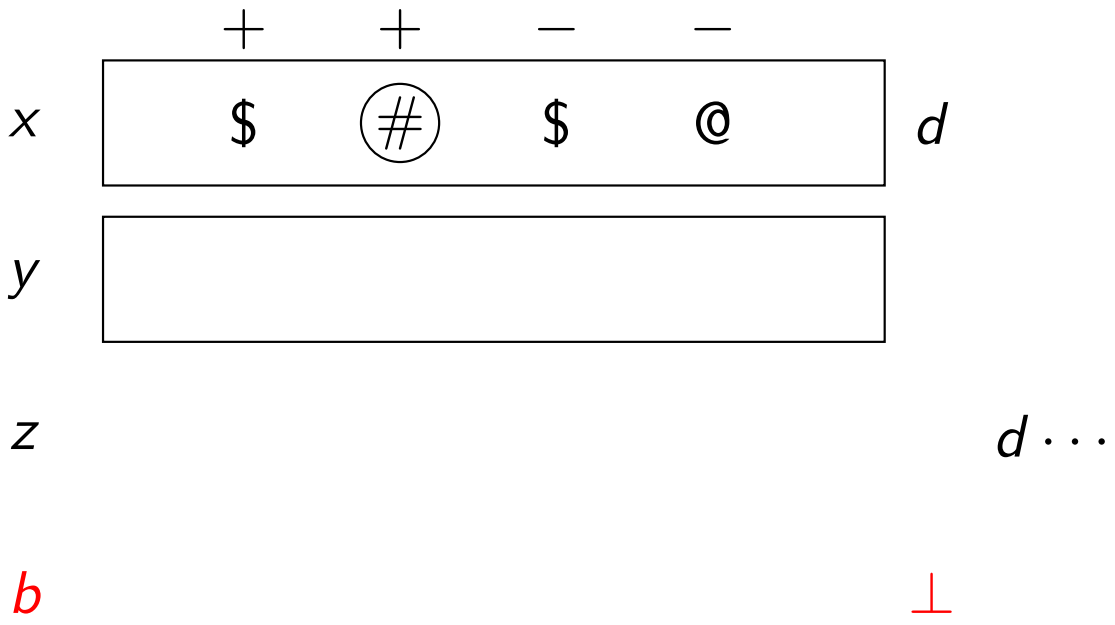
Simulating counter machines



Increment

- d should be a new data value.
- If not, second player can set b to false.

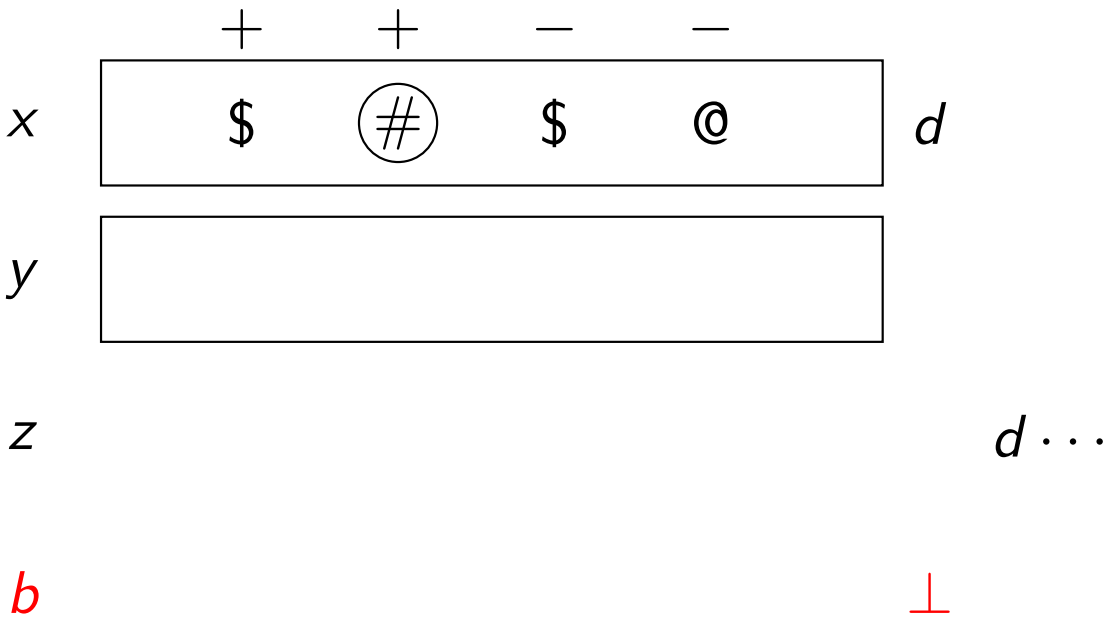
Simulating counter machines



Increment

- d should be a new data value.
- If not, second player can set b to false.
- First player must repeat d in z from next round.

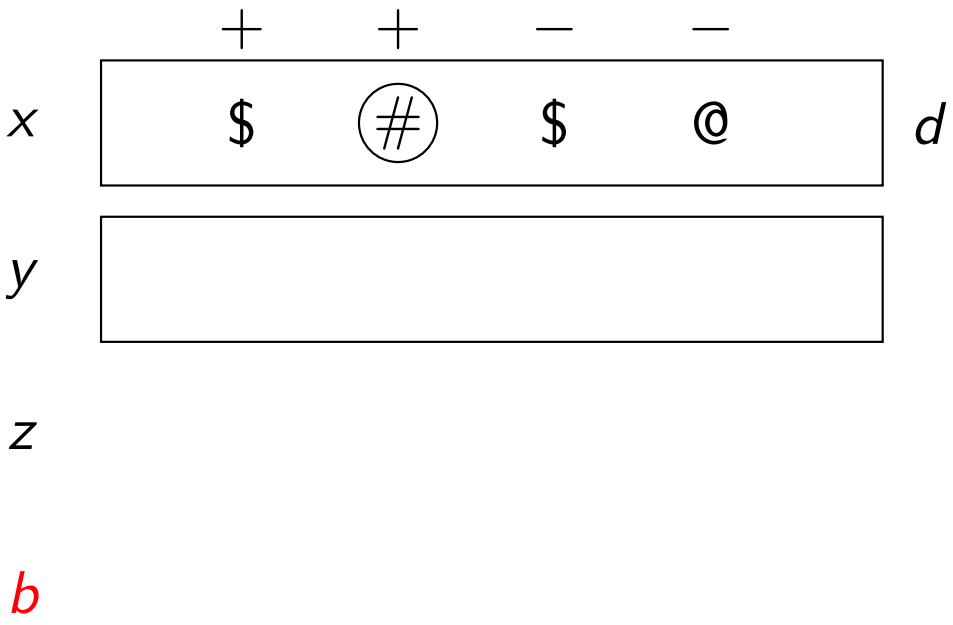
Simulating counter machines



Increment

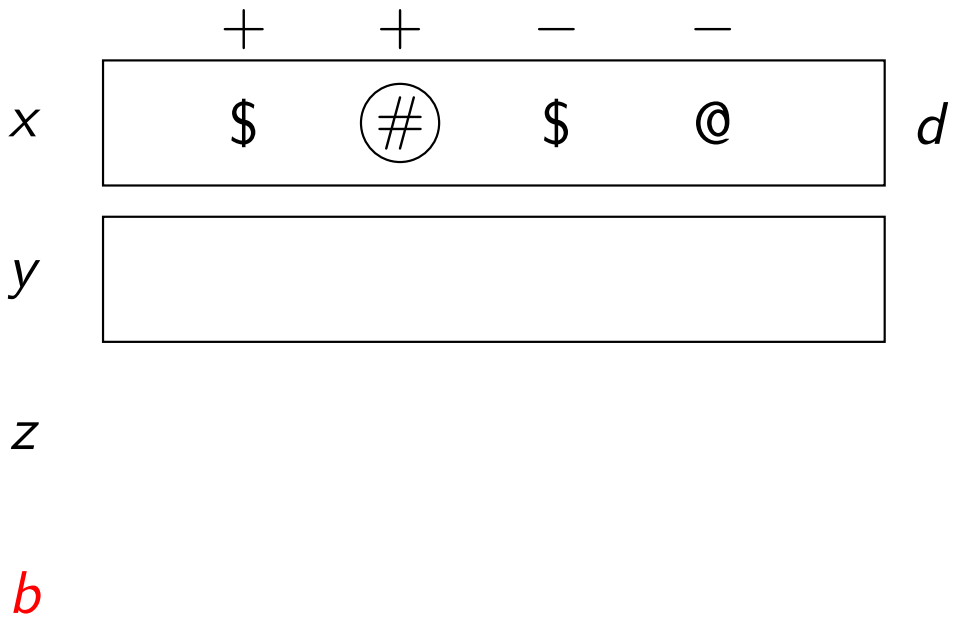
- d should be a new data value.
- If not, second player can set b to false.
- First player must repeat d in z from next round.
- If cheating on increment, $G(b \Rightarrow \neg(x \approx \Diamond z))$

Simulating counter machines



Decrement

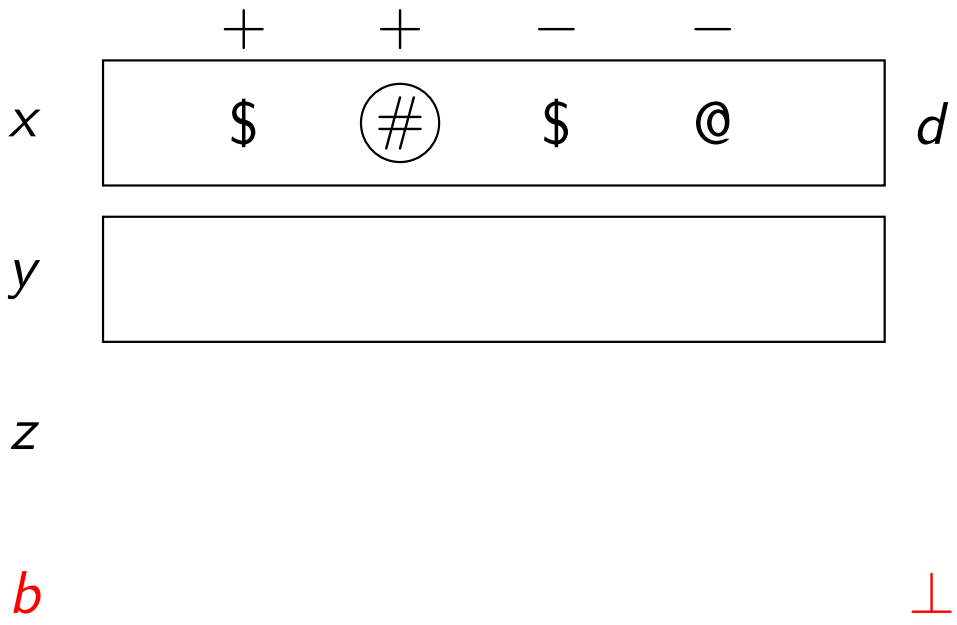
Simulating counter machines



Decrement

- d must repeat in the past in an incrementing position.

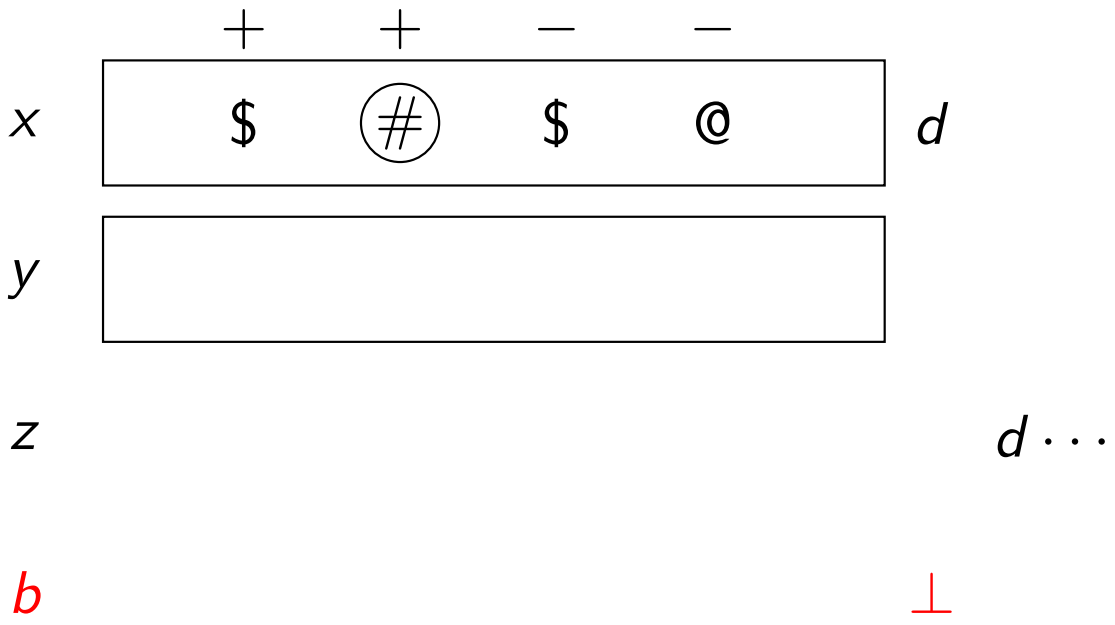
Simulating counter machines



Decrement

- d must repeat in the past in an incrementing position.
- If not, second player can set b to false.

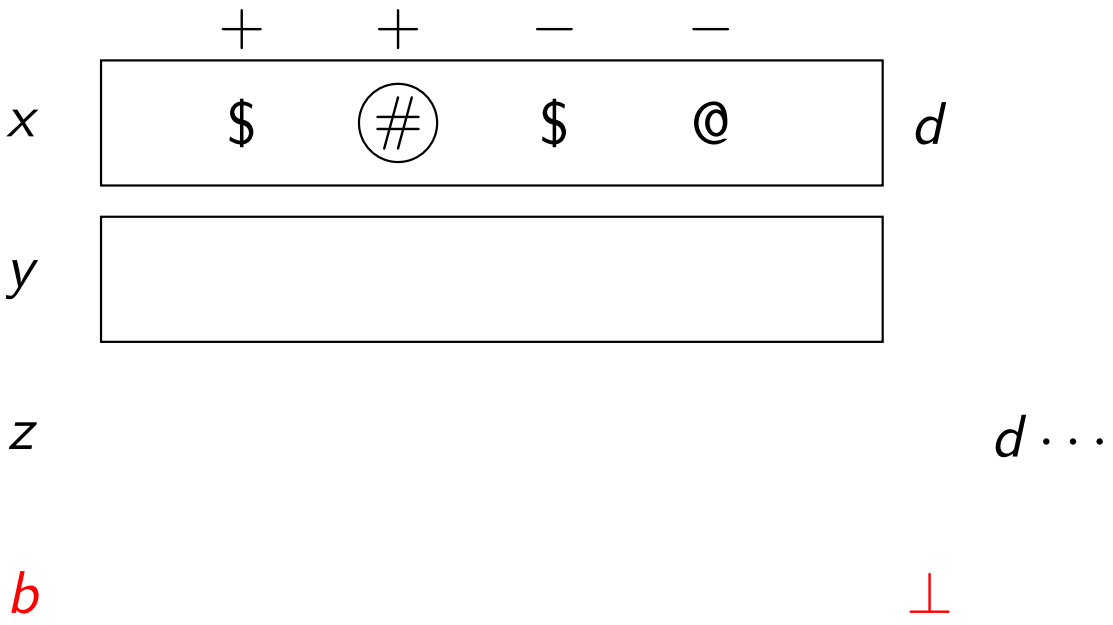
Simulating counter machines



Decrement

- d must repeat in the past in an incrementing position.
- If not, second player can set b to false.
- First player must repeat d in z from next round.

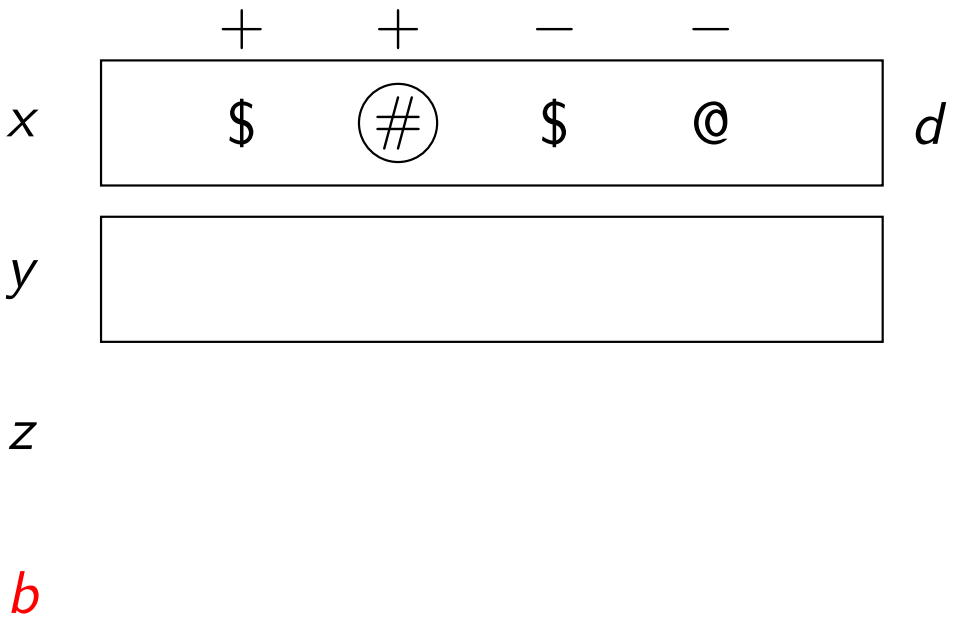
Simulating counter machines



Decrement

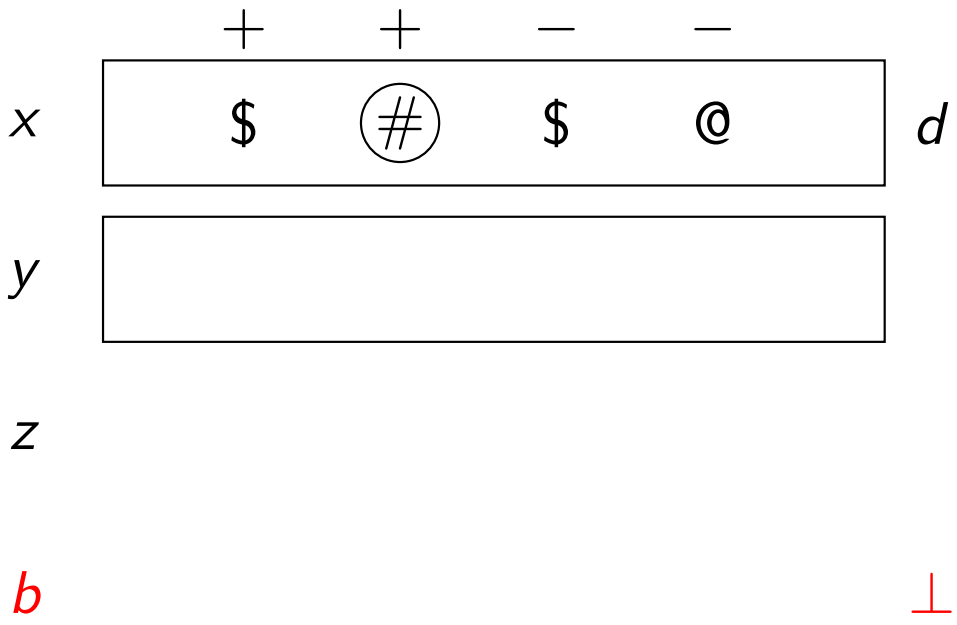
- d must repeat in the past in an incrementing position.
- If not, second player can set b to false.
- First player must repeat d in z from next round.
- If cheating on decrement, $F(b \wedge x \approx \Diamond z)$.

Simulating counter machines



Zero test

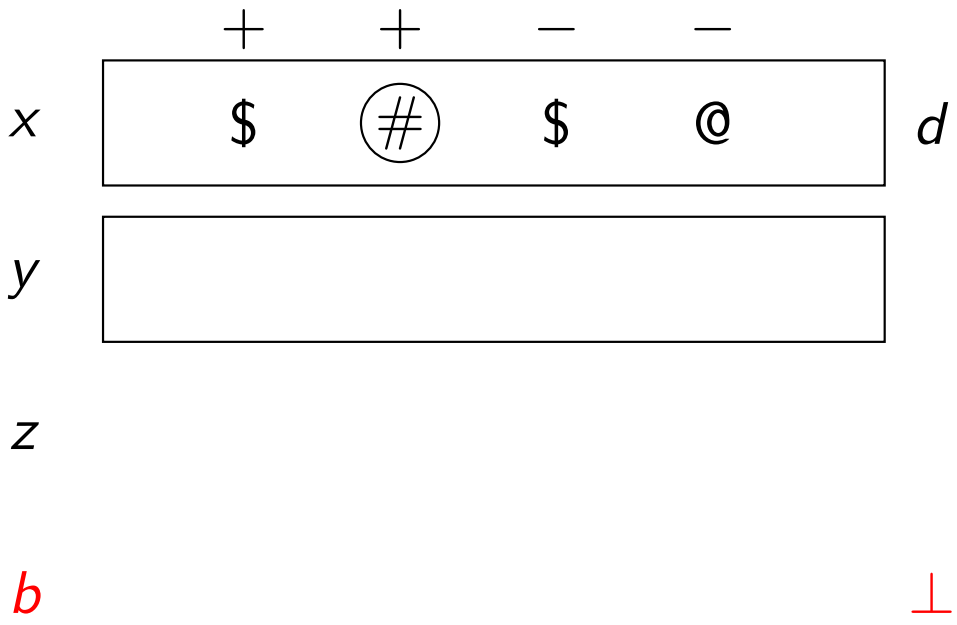
Simulating counter machines



Zero test

- Increment, no matching decrement $\Rightarrow$ b set to false.

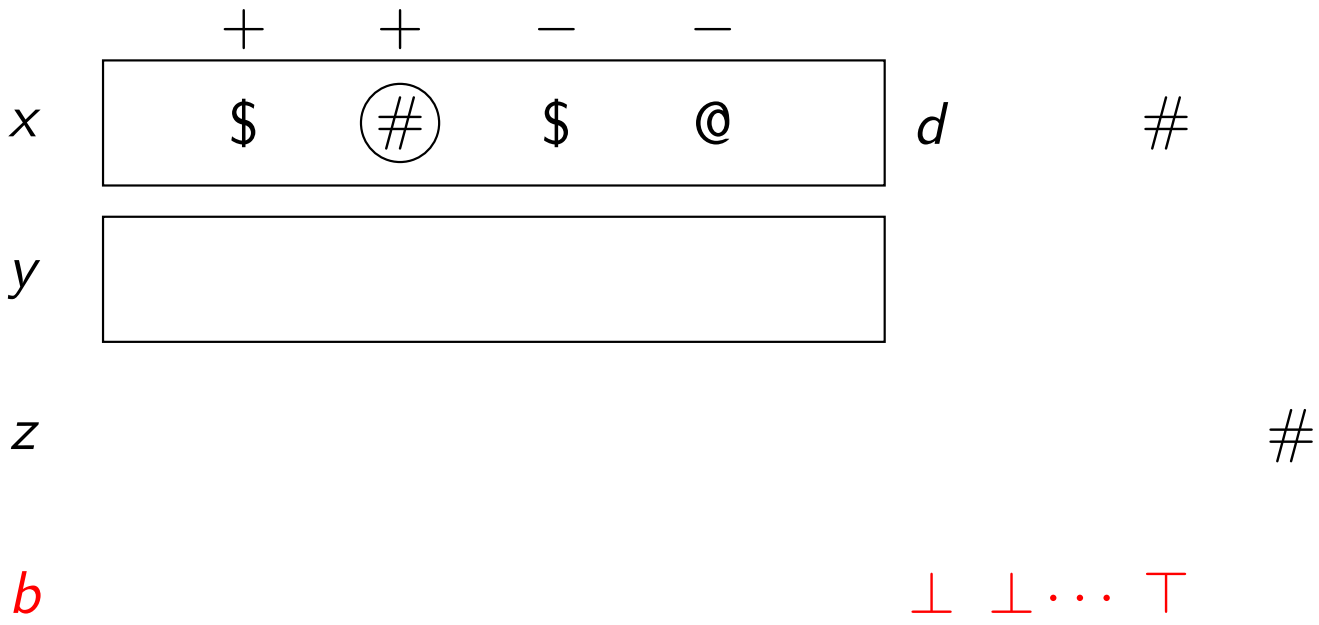
Simulating counter machines



Zero test

- Increment, no matching decrement $\Rightarrow$ b set to false.
- Increment, future zero test $\Rightarrow$ x must repeat in x.

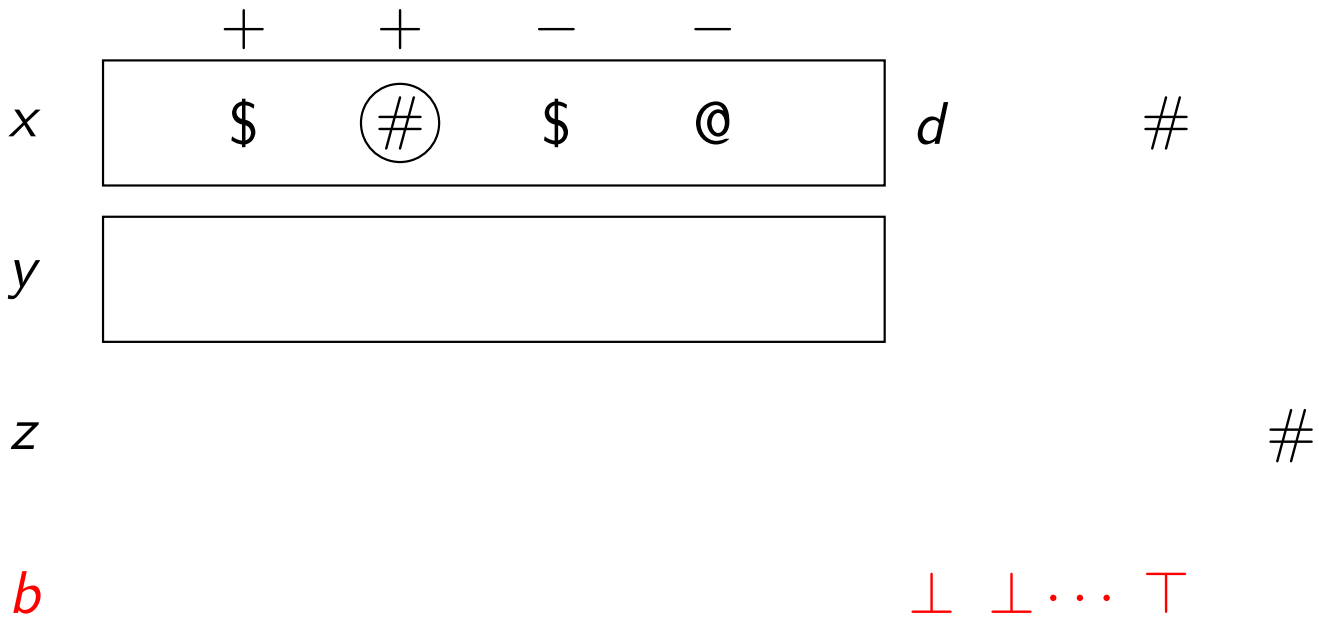
Simulating counter machines



Zero test

- Increment, no matching decrement $\Rightarrow$ b set to false.
- Increment, future zero test $\Rightarrow$ x must repeat in x.
- Upon repetition, b is set to true again; first player must repeat $\#$ in z.

Simulating counter machines



Zero test

- Increment, no matching decrement $\Rightarrow$ b set to false.
- Increment, future zero test $\Rightarrow$ x must repeat in x.
- Upon repetition, b is set to true again; first player must repeat $\#$ in z.
- If cheating on zero test Globally (b is true and incrementing $\Rightarrow$ x doesn't repeat in z).

Outline

Restrictions to get decidability

- No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

Restrictions to get decidability

- ~~No nested formulas: only $x \approx \langle \top ? \rangle^{-1} y$.~~
- Environment player has only Boolean variables.
- No future obligations: $x \approx \langle \top ? \rangle y$ not allowed.

Future work

- Complexity bounds.
- Synthesizing winning strategies.
- Other decidable restrictions of VASS games.

Future work

- Complexity bounds.
- Synthesizing winning strategies.
- Other decidable restrictions of VASS games.

Thank you