UNIVERSITÉ LIBRE DE BRUXELLES

Robustness and Implementability of Timed Automata

FORMATS-FTRTFT 2004 – Sep 24th - Grenoble

Martin De Wulf

Laurent Doyen

Nicolas Markey

Jean-François Raskin

Centre Fédéré en Vérification

Motivation

- Embedded Controllers

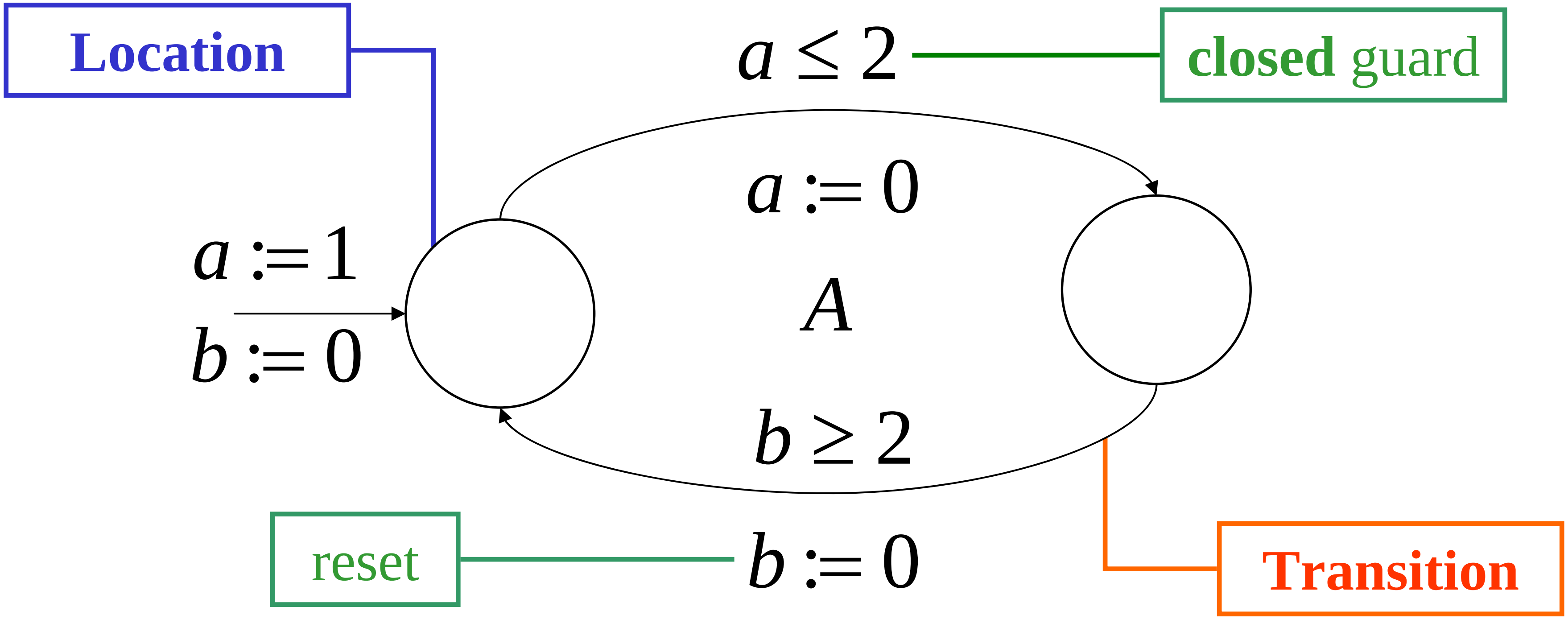
  … are difficult to develop (concurrency, real-time, continuous environment, …).

  … are safety critical.

➡ Use formal models + Verification:

Timed Automata and
Reachability Analysis

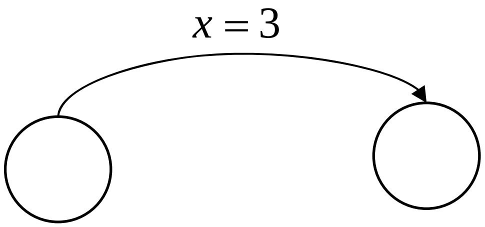
Timed Automata



Clocks: {a,b}

Objectives

- From a *verified* model, generate (automatically) a *correct* implementation
- Using classical formalism (e.g. timed automata)
- …but interpreting the model in a way that guarantees the transfer of the properties from model to implementation



ULB UNIVERSITÉ LIBRE DE BRUXELLES

Robustness and Implentation

Model vs. Implementation

- Perfect continuous clocks
- Instantaneous synchronisations
- Reaction time = 0

- Digital clocks
- Delayed synchronisations
- Reaction time > 0

Correct model → ? → Correct implementation

Timed Automata: Semantics

Classical $[A]_0^0$ vs. Enlarged $[A]_\Delta^\varepsilon$

Perfect clocks

Rate: $dx/dt = 1$

Guard: $v \models g \text{ iff } v \in [g]$

$[a \le x \le b] = [a, b]$

Imprecise clocks

Rate: $dx/dt \in [1-\varepsilon, 1+\varepsilon]$

Guard: $v \models g \text{ iff } v \in [g]_\Delta$

$[a \le x \le b]_\Delta = [a-\Delta, b+\Delta]$

Shortcut: $[A]^\varepsilon := [A]_{\Delta=0}^\varepsilon \qquad [A]_\Delta := [A]_\Delta^{\varepsilon=0}$

From Model to Implementation

Timed automaton A is **correct** if

$$\mathrm{Reach}([A]) \cap \mathrm{Bad} = \emptyset$$

Timed automaton A is **implementable** [DDR04] if

$$\exists \Delta > 0\ \mathrm{Reach}([A']_\Delta) \cap \mathrm{Bad} = \emptyset$$

which is equivalent to

$$\bigcap_{\Delta > 0} \mathrm{Reach}([A']_\Delta) \cap \mathrm{Bad} = \emptyset$$

Robustness

- Is it always the case that $\bigcap_{\Delta>0} \mathrm{Reach}([A]_\Delta) = \mathrm{Reach}([A])$?

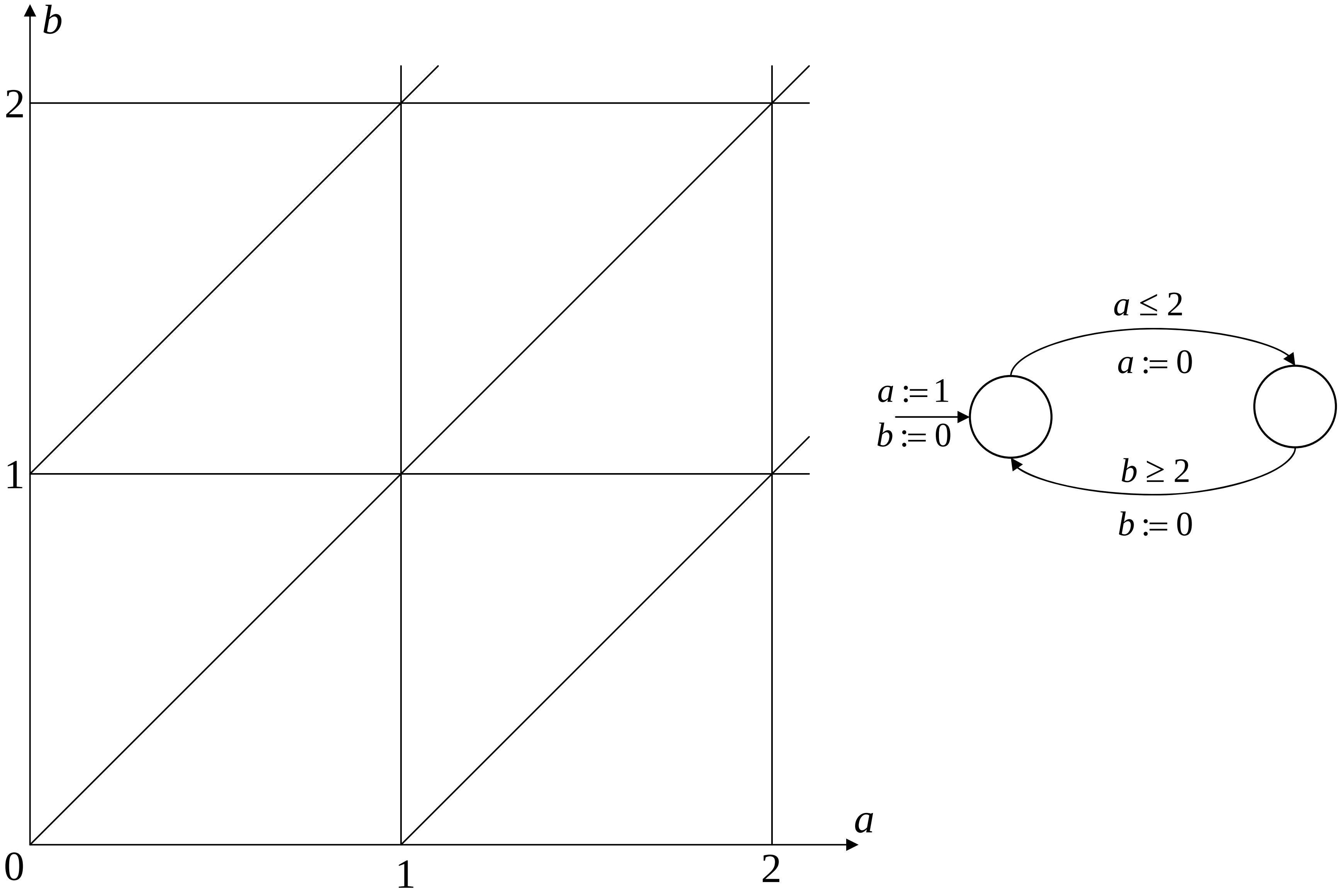
  No ! (see example)

- How to compute $\bigcap_{\Delta>0} \mathrm{Reach}([A]_\Delta)$?
  - [Pur98] gives an algorithm for $\bigcap_{\varepsilon>0} \mathrm{Reach}([A]^\varepsilon)$
  - We show that $\bigcap_{\varepsilon>0} \mathrm{Reach}([A]^\varepsilon) = \bigcap_{\Delta>0} \mathrm{Reach}([A]_\Delta)$

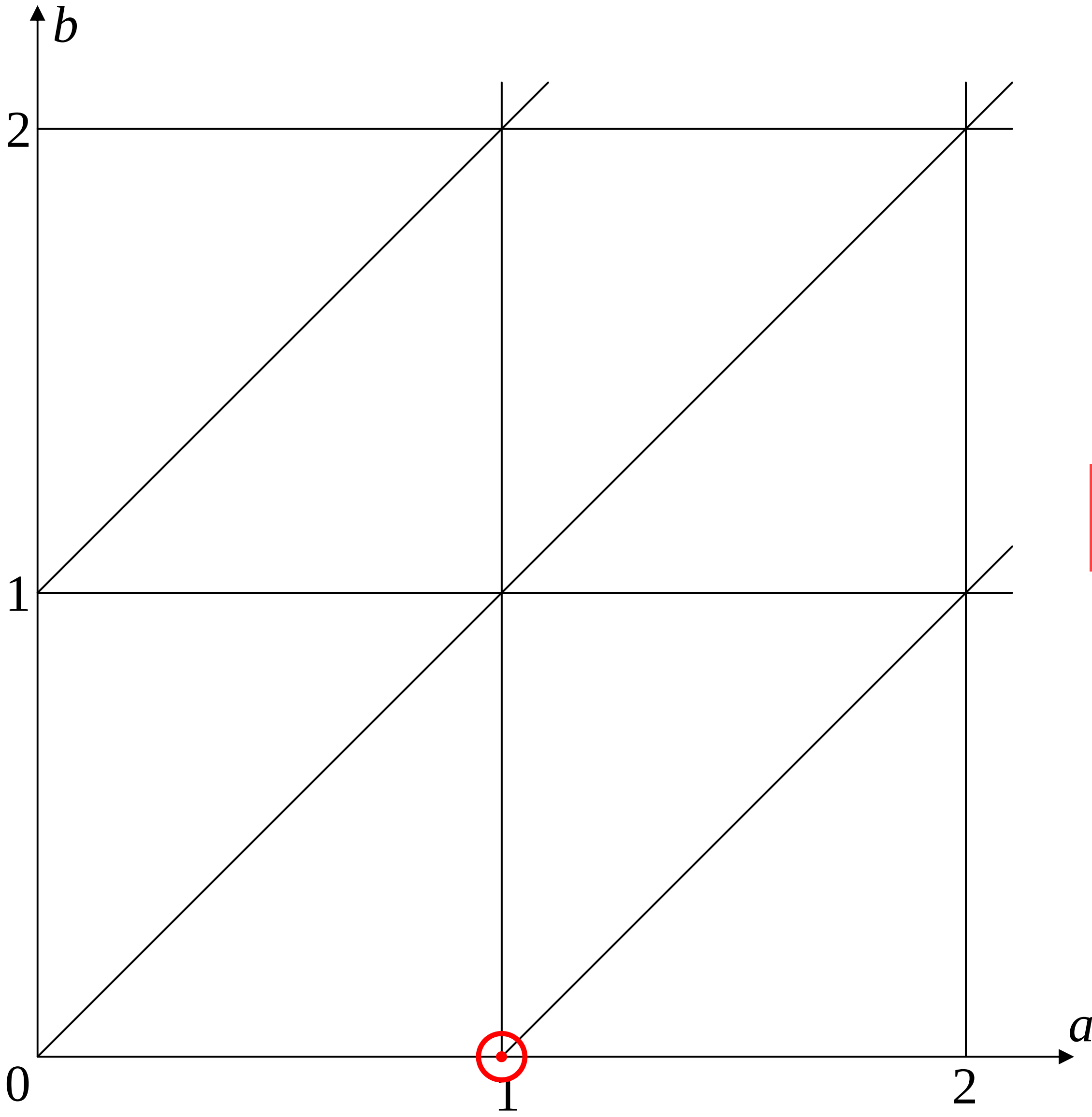
An example showing that

$$\mathrm{Reach}([A]) \neq \bigcap_{\Delta>0} \mathrm{Reach}([A]_\Delta)$$

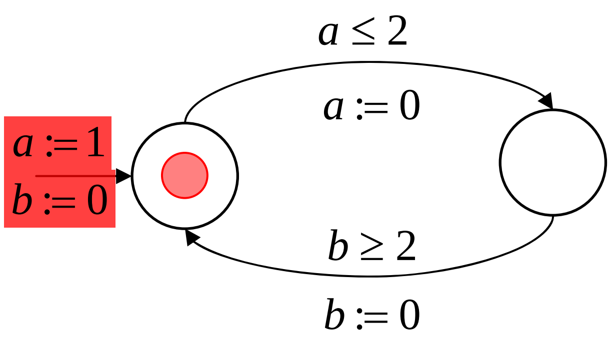
Example



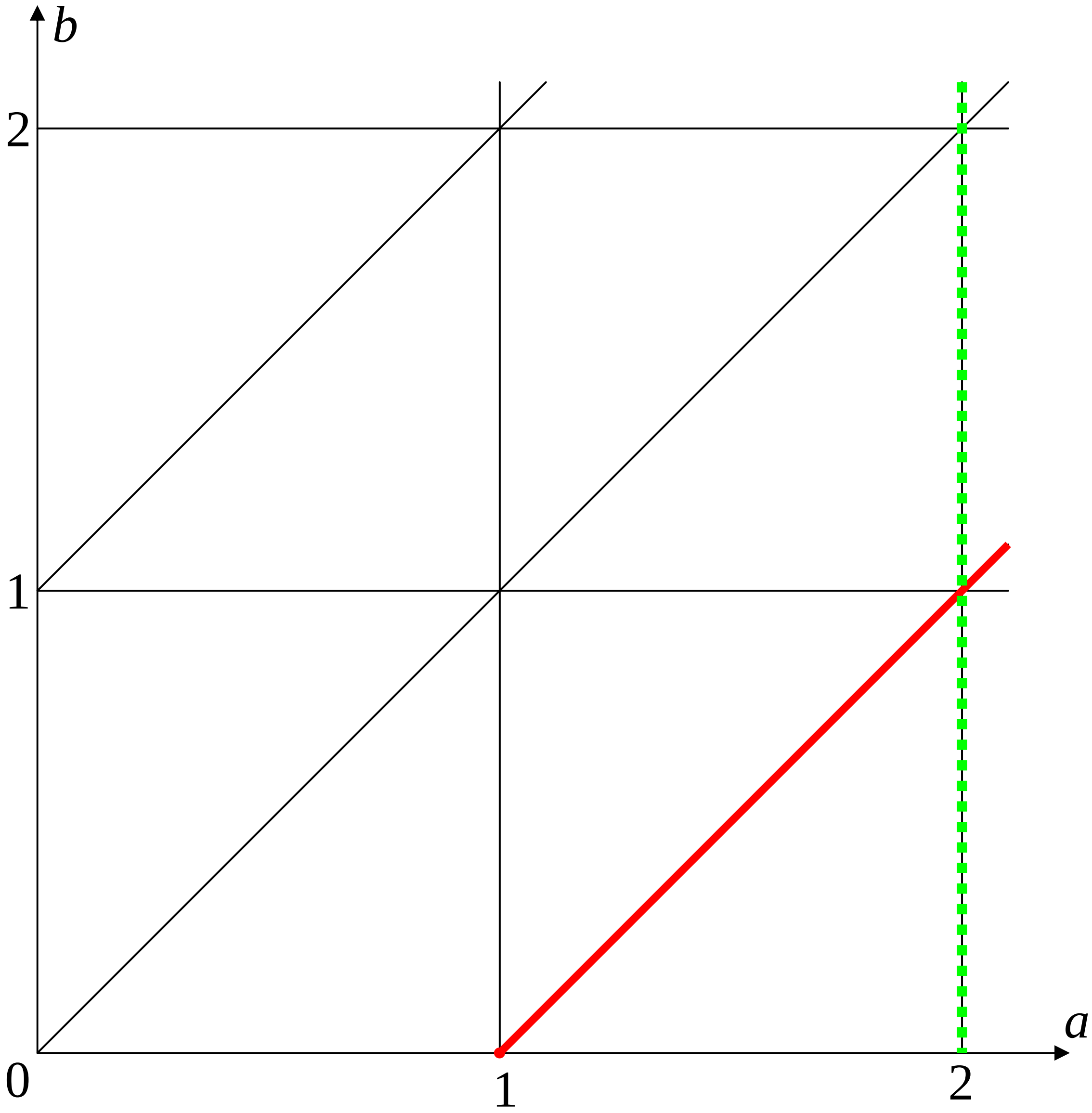
Example



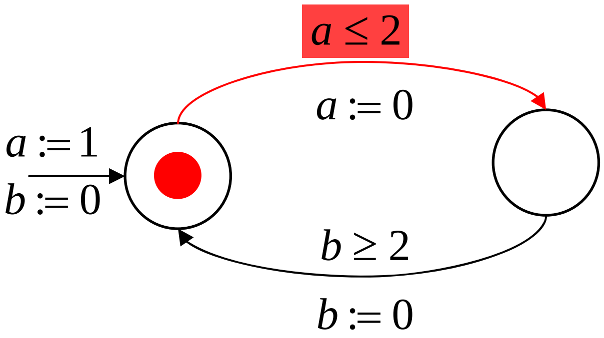
Classical Semantics



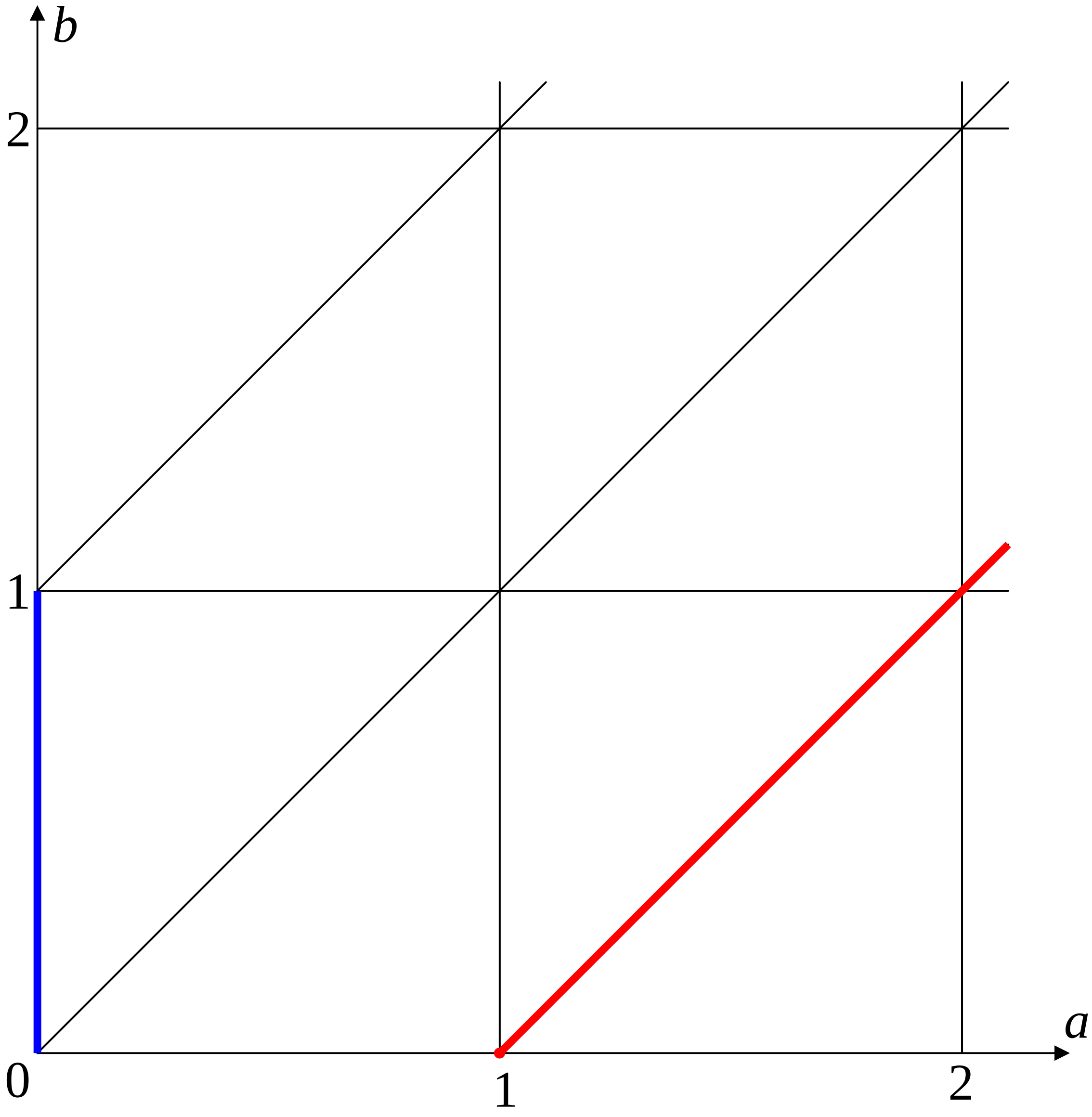
Example



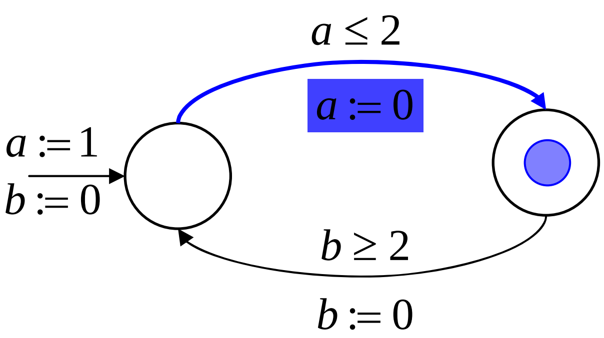
Classical Semantics



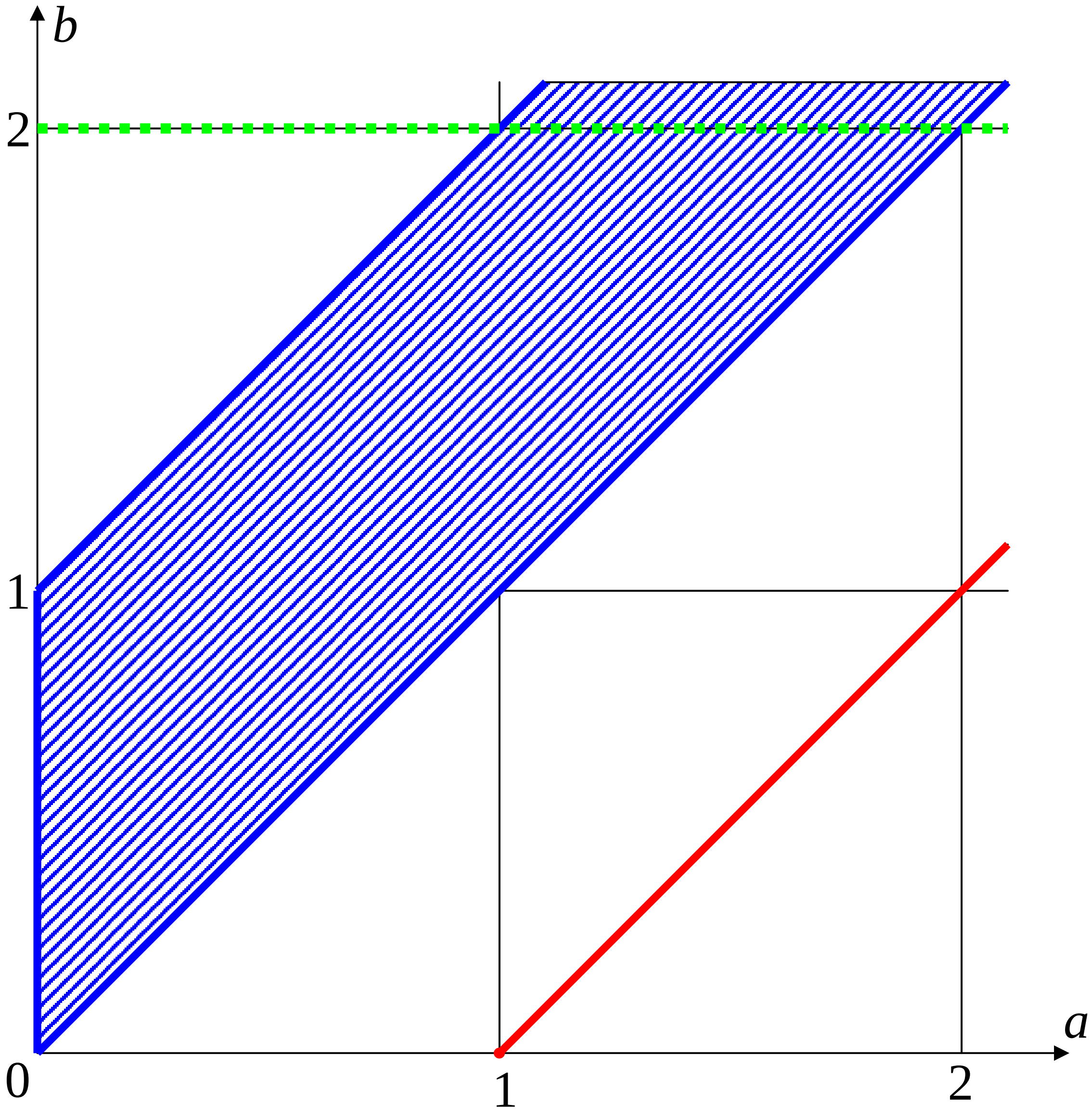
Example



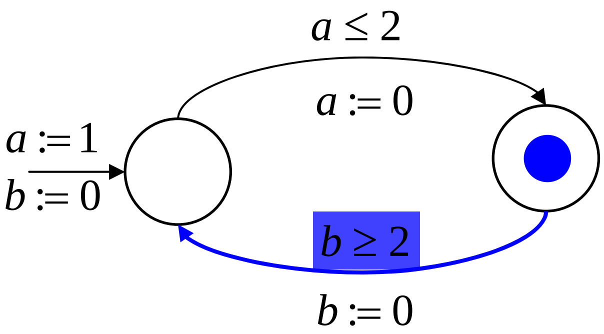
Classical Semantics



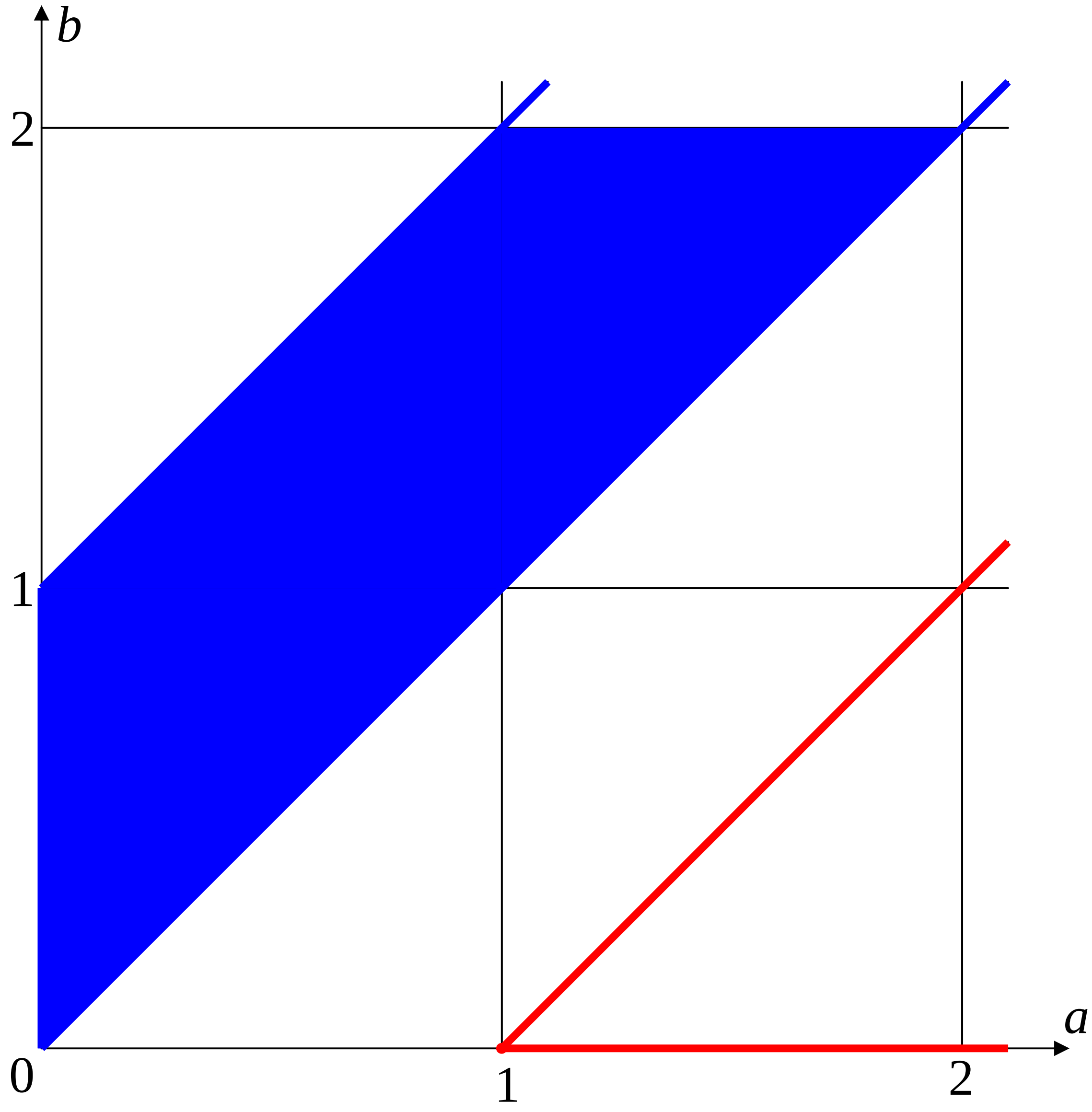
Example



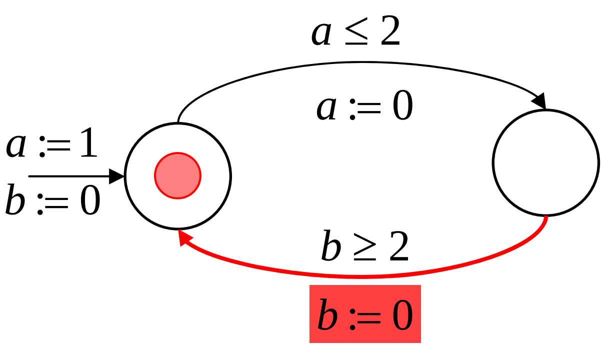
Classical Semantics



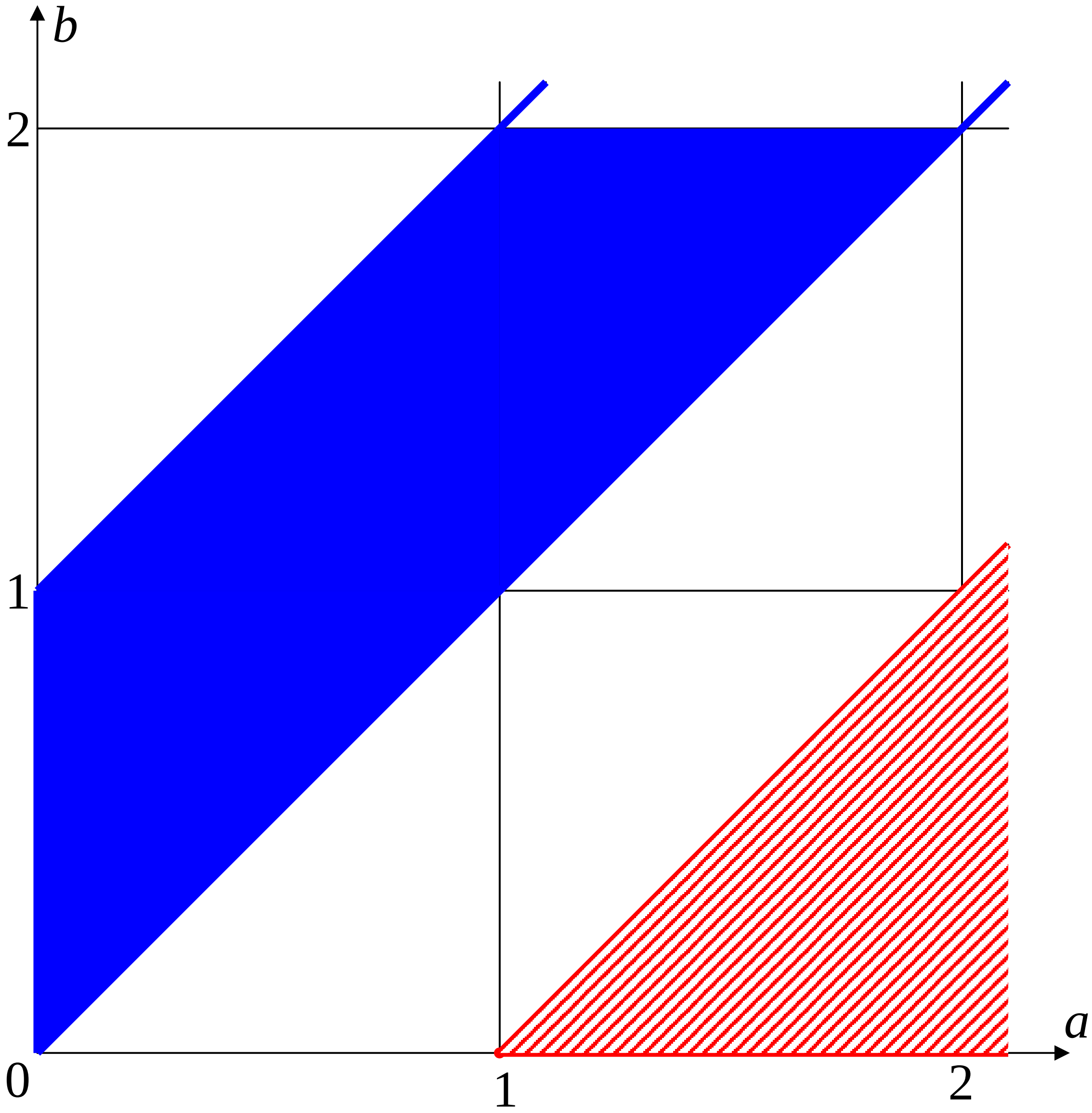
Example



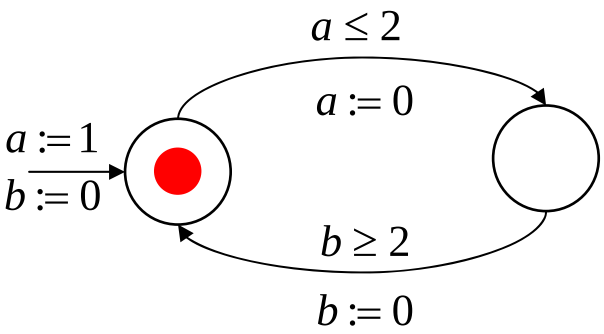
Classical Semantics



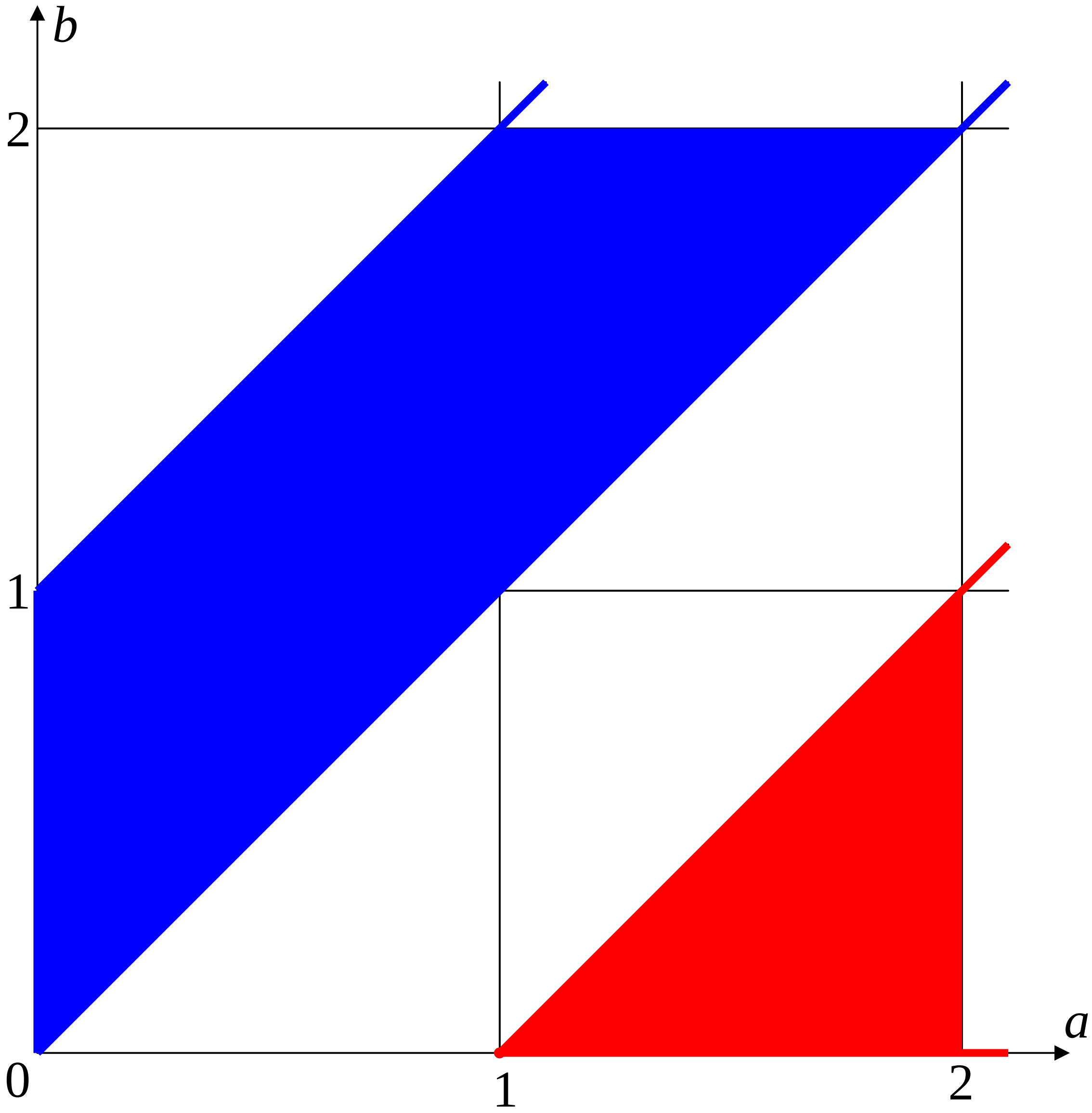
Example



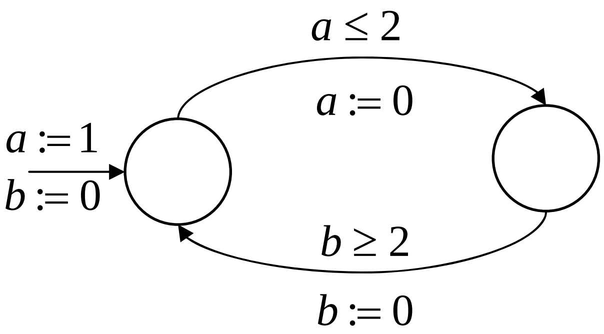
Classical Semantics



Example

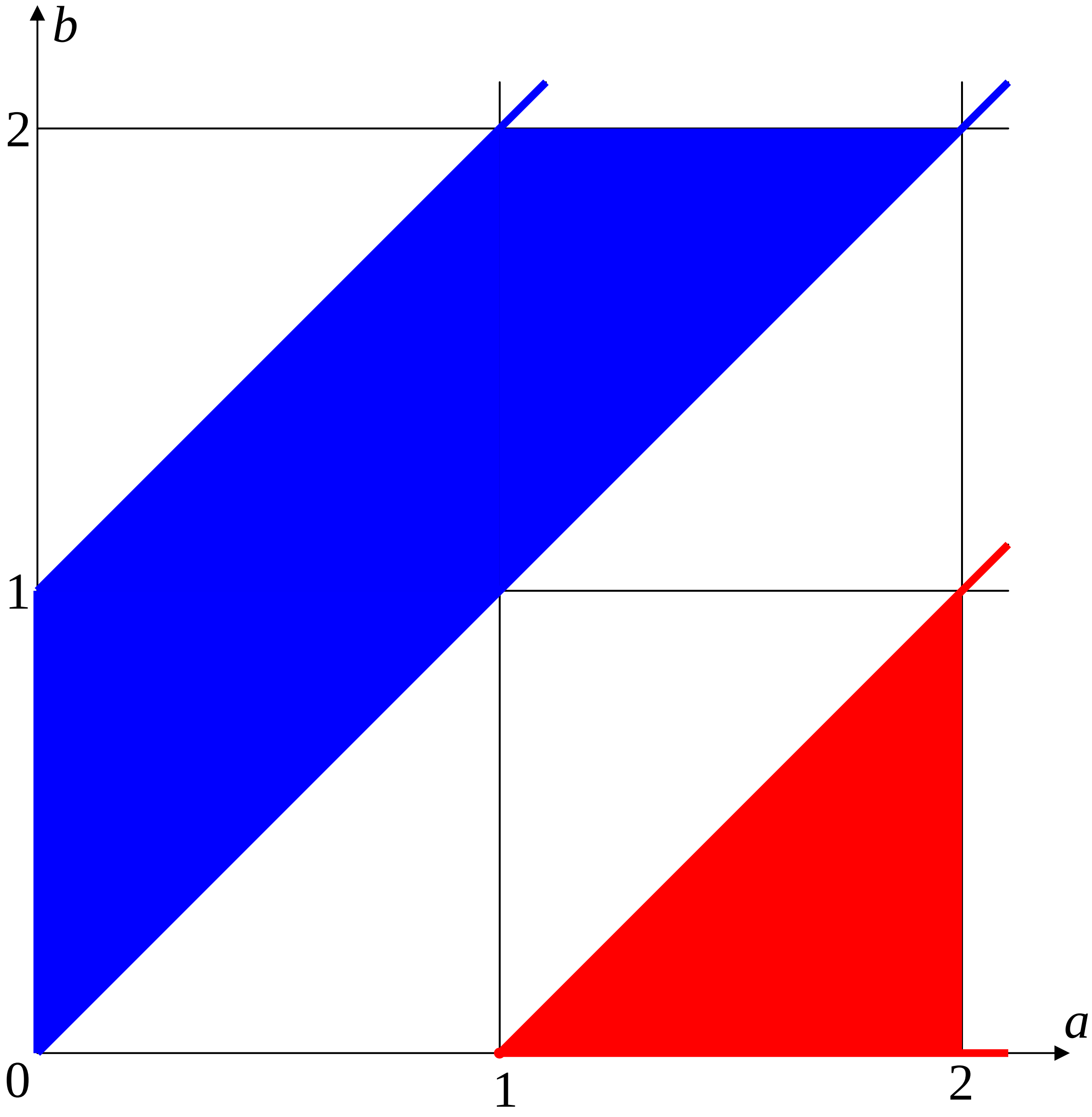


Classical Semantics

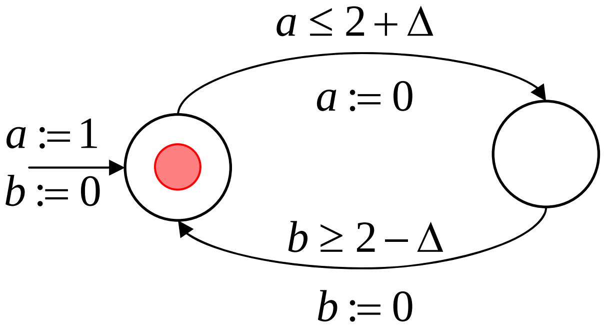


$\mathsf{Reach}([A])$

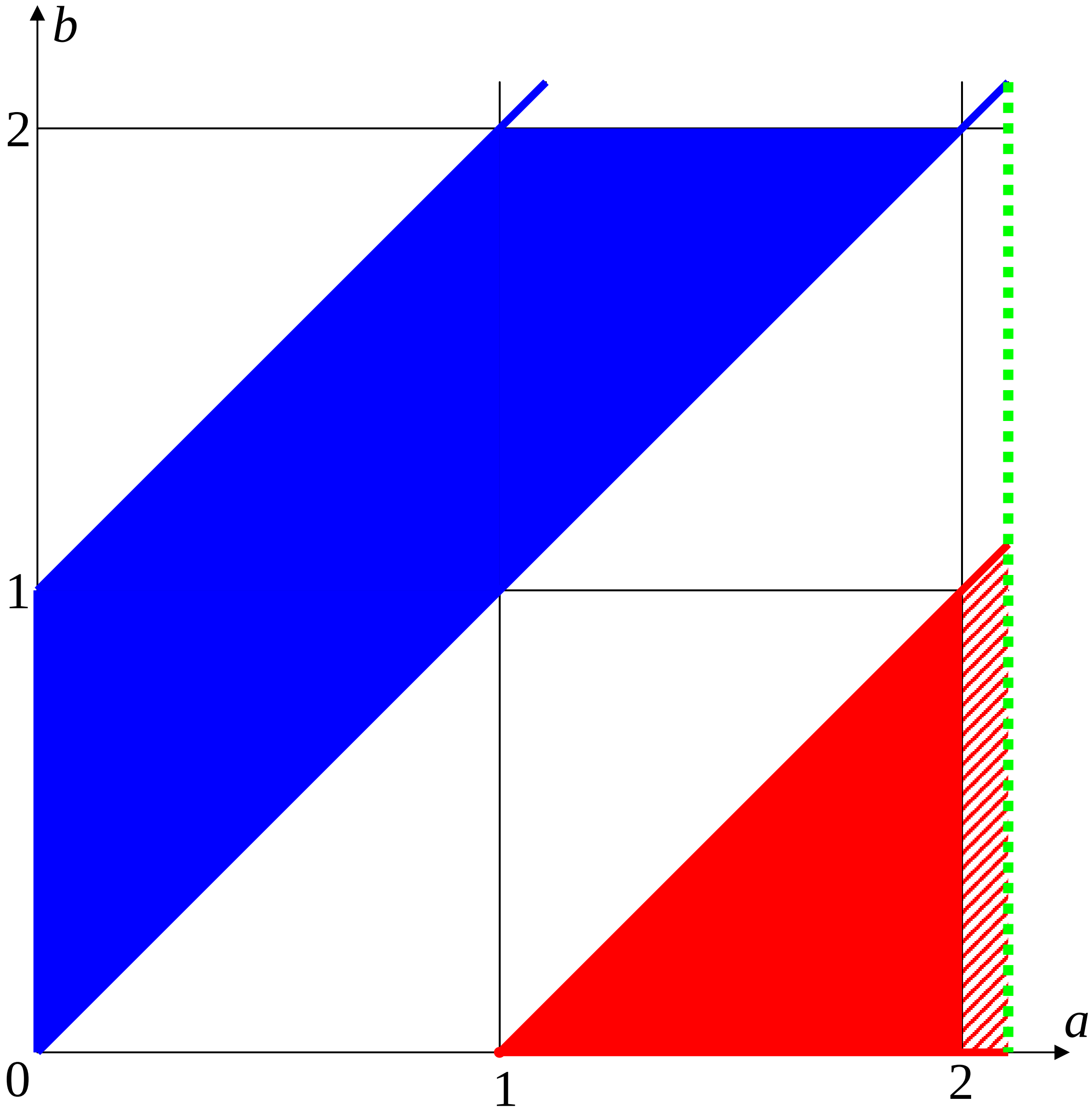
Example



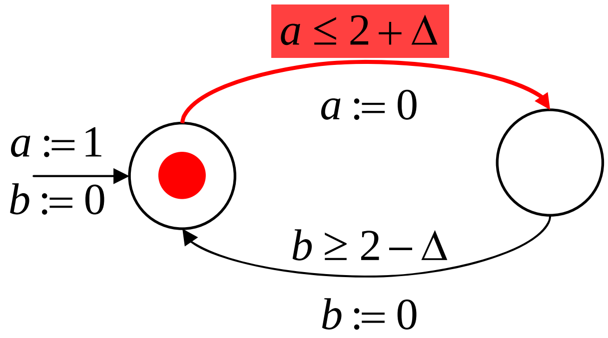
Enlarged Semantics



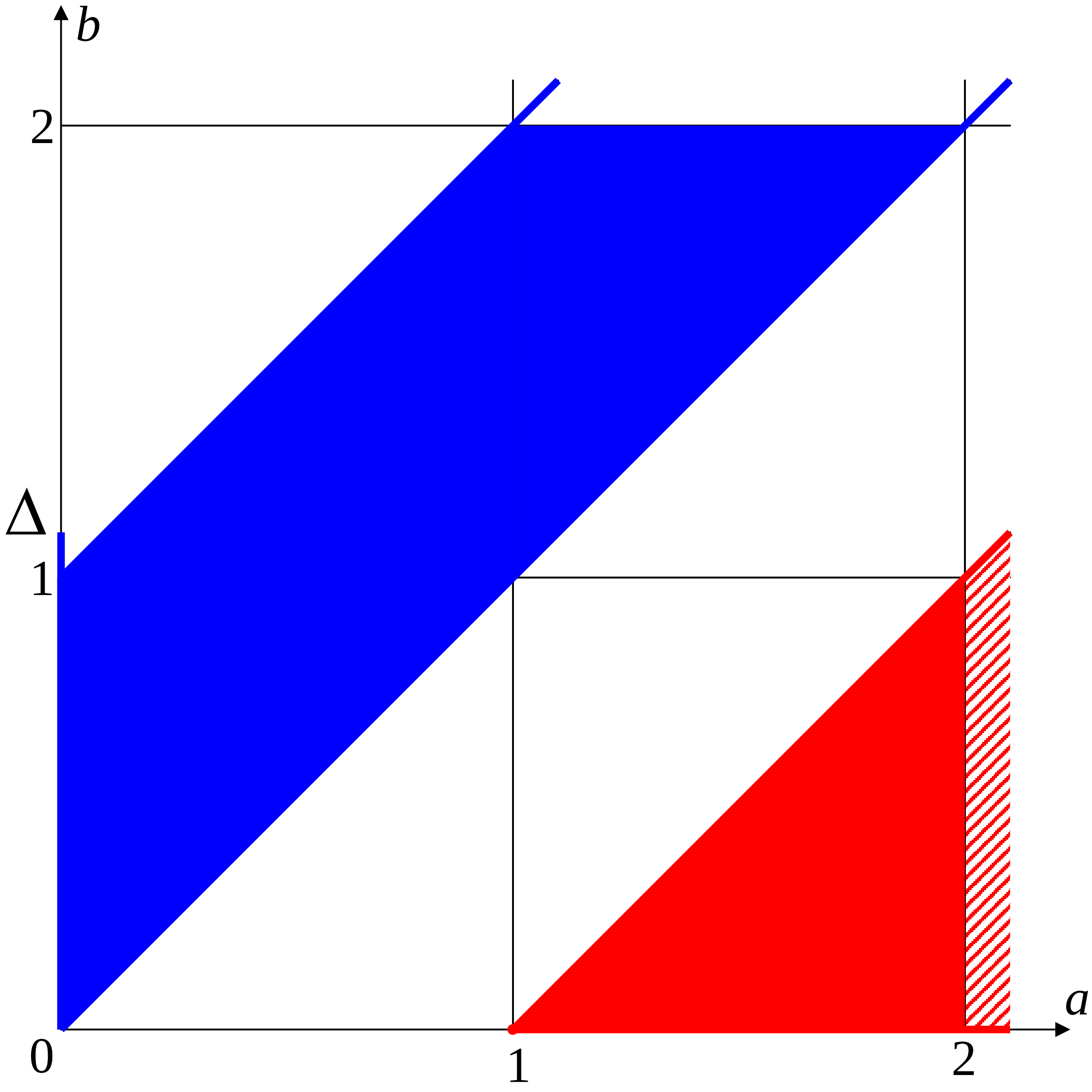
Example



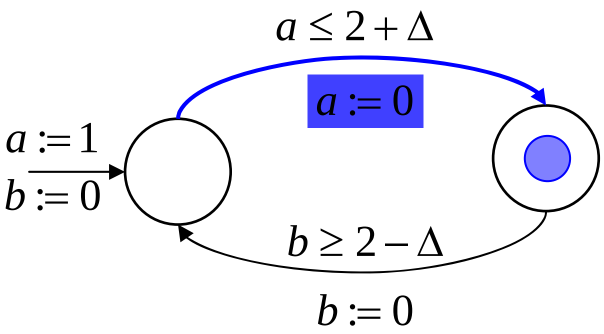
Enlarged Semantics



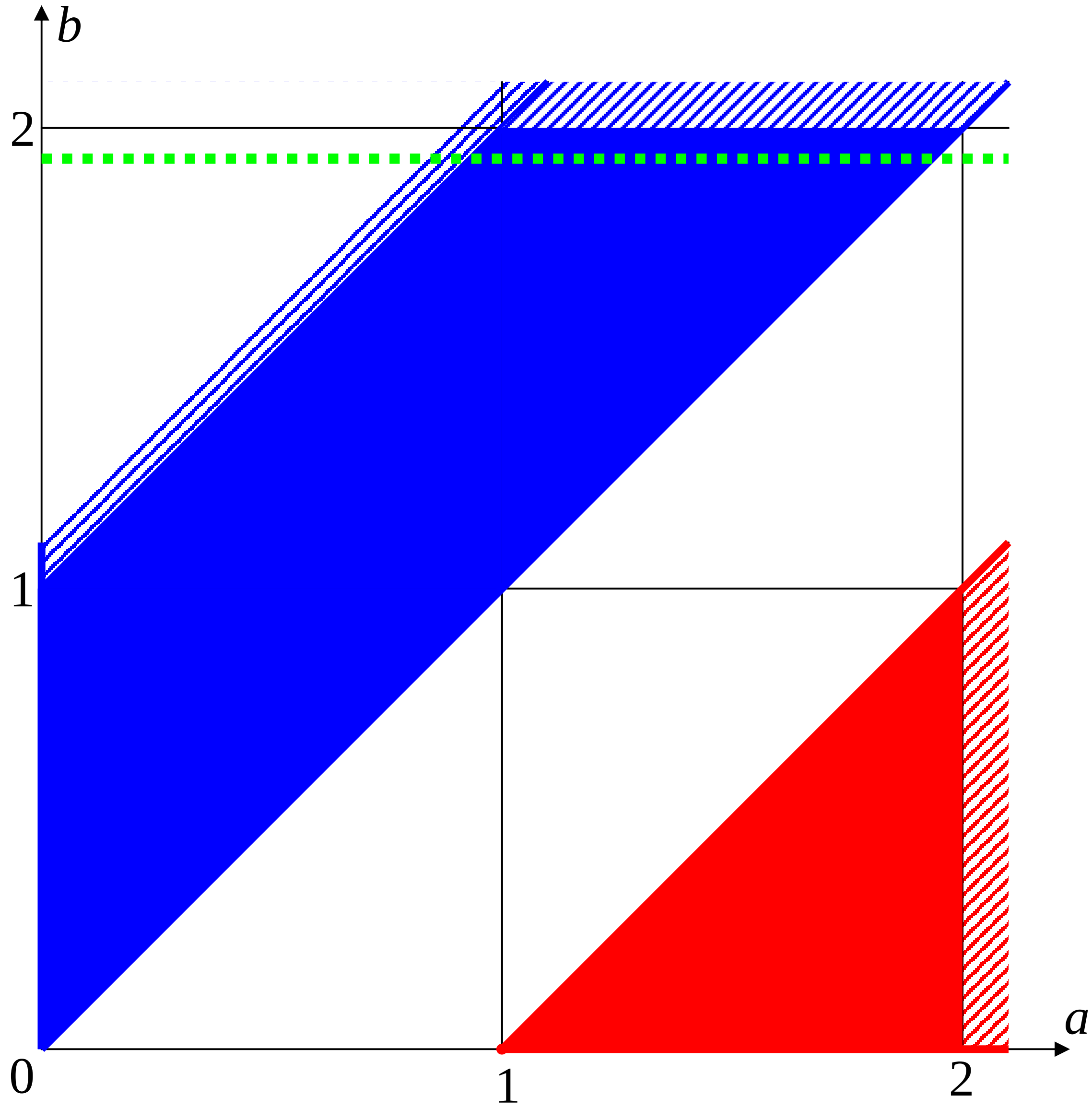
Example



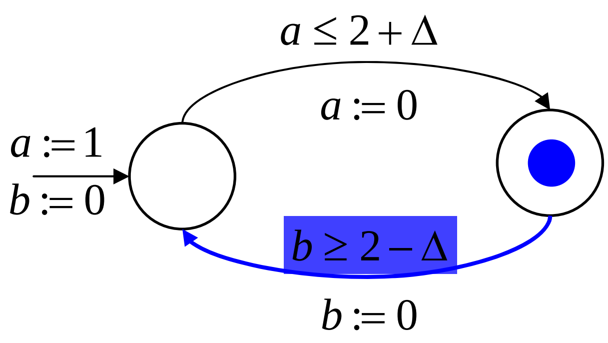
Enlarged Semantics



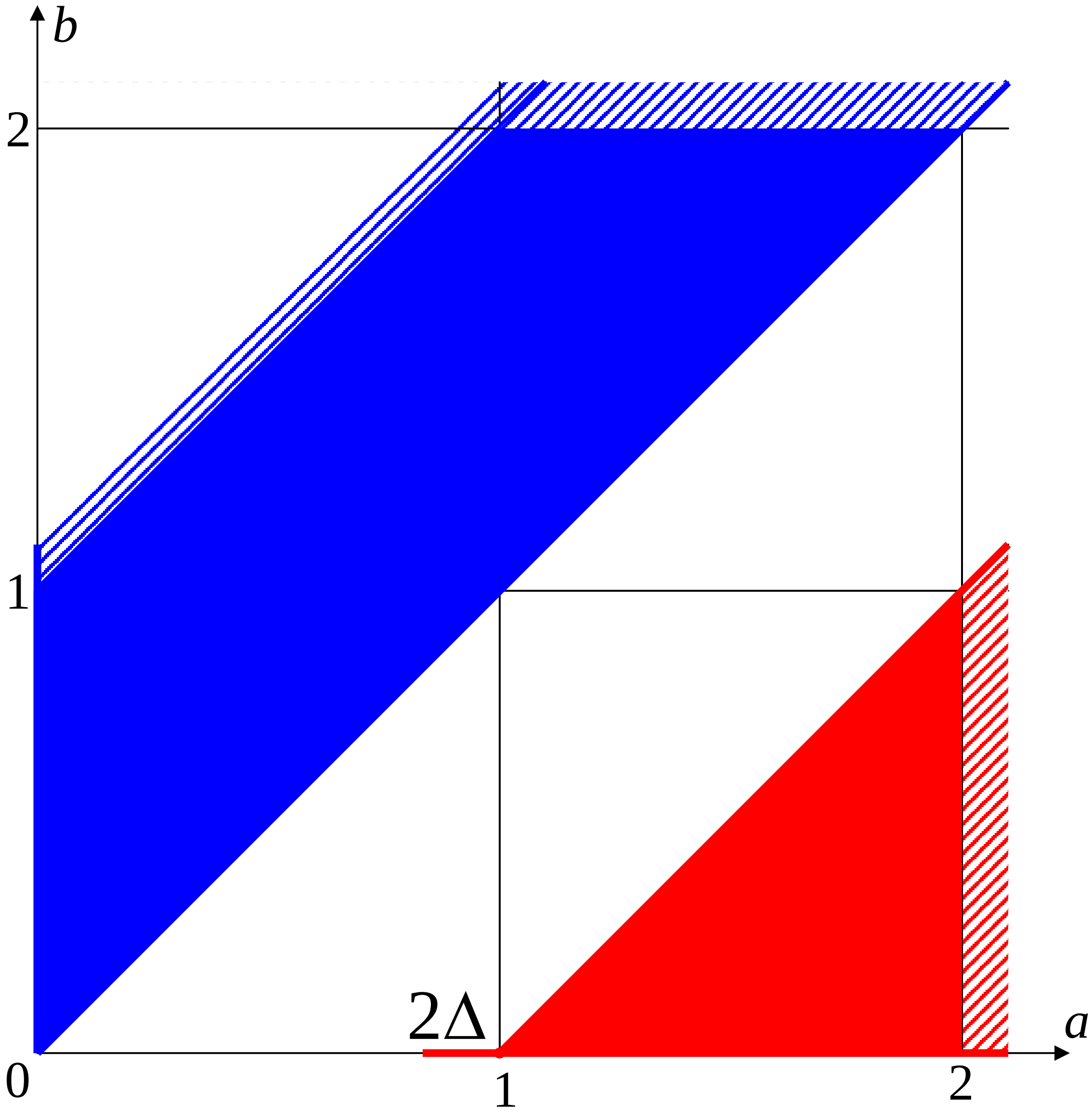
Example



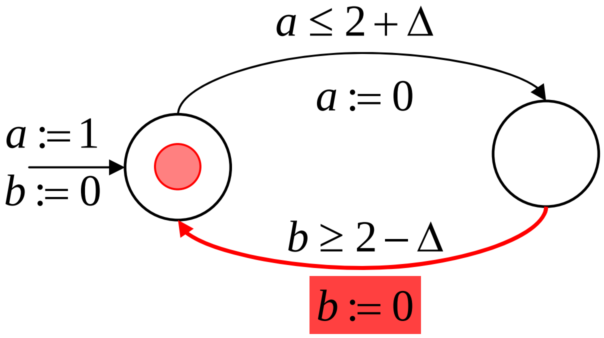
Enlarged Semantics



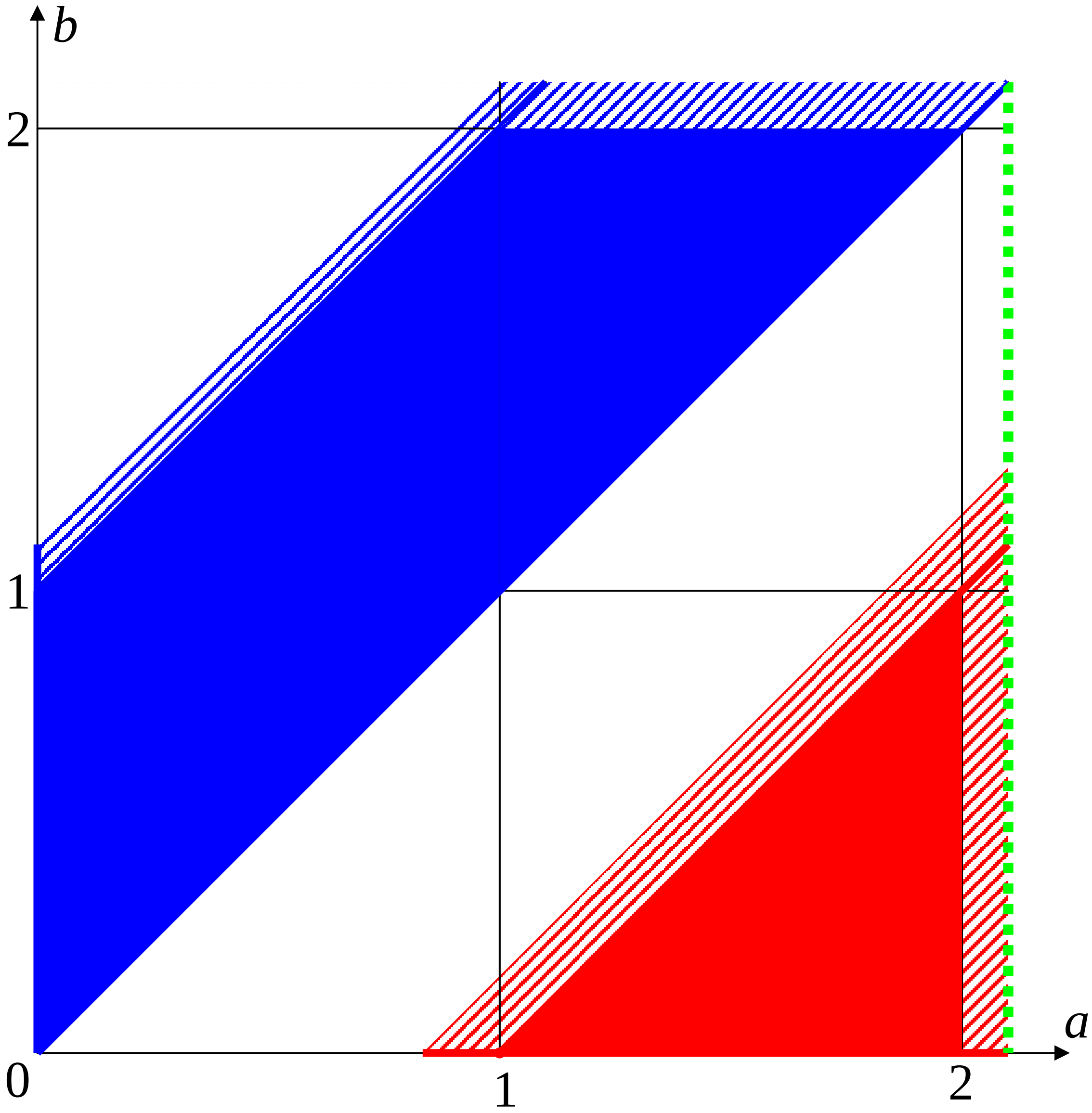
Example



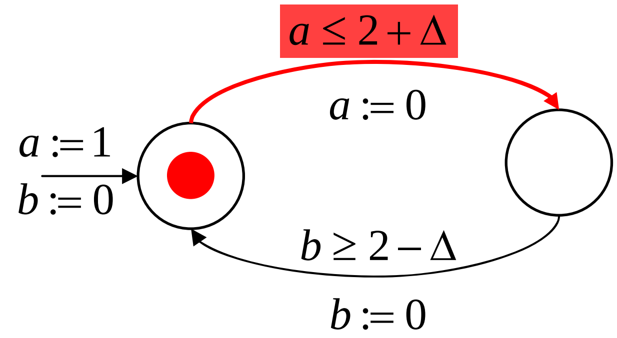
Enlarged Semantics



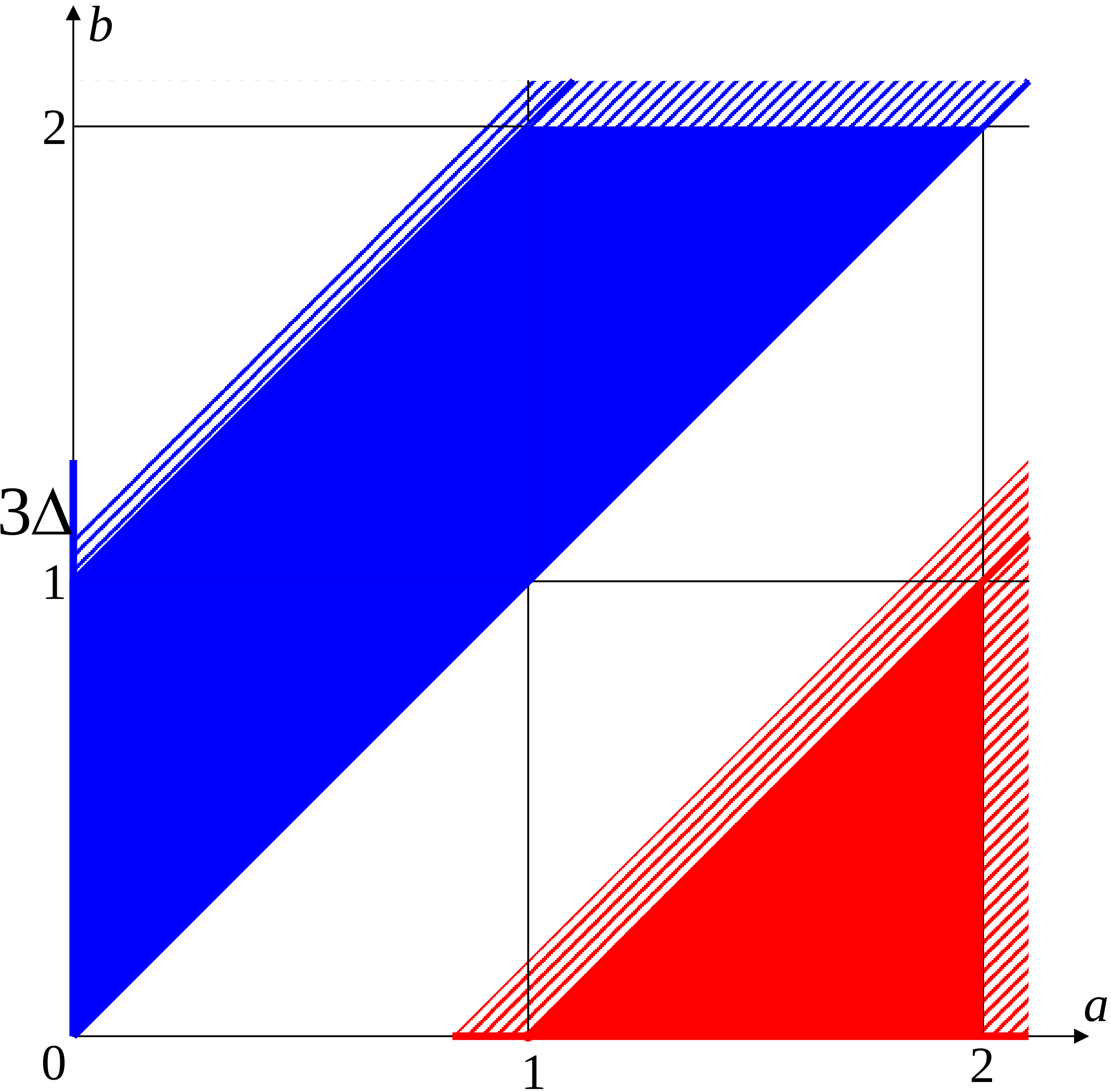
Example



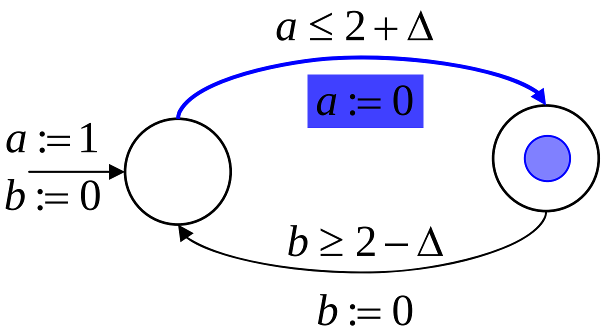
Enlarged Semantics



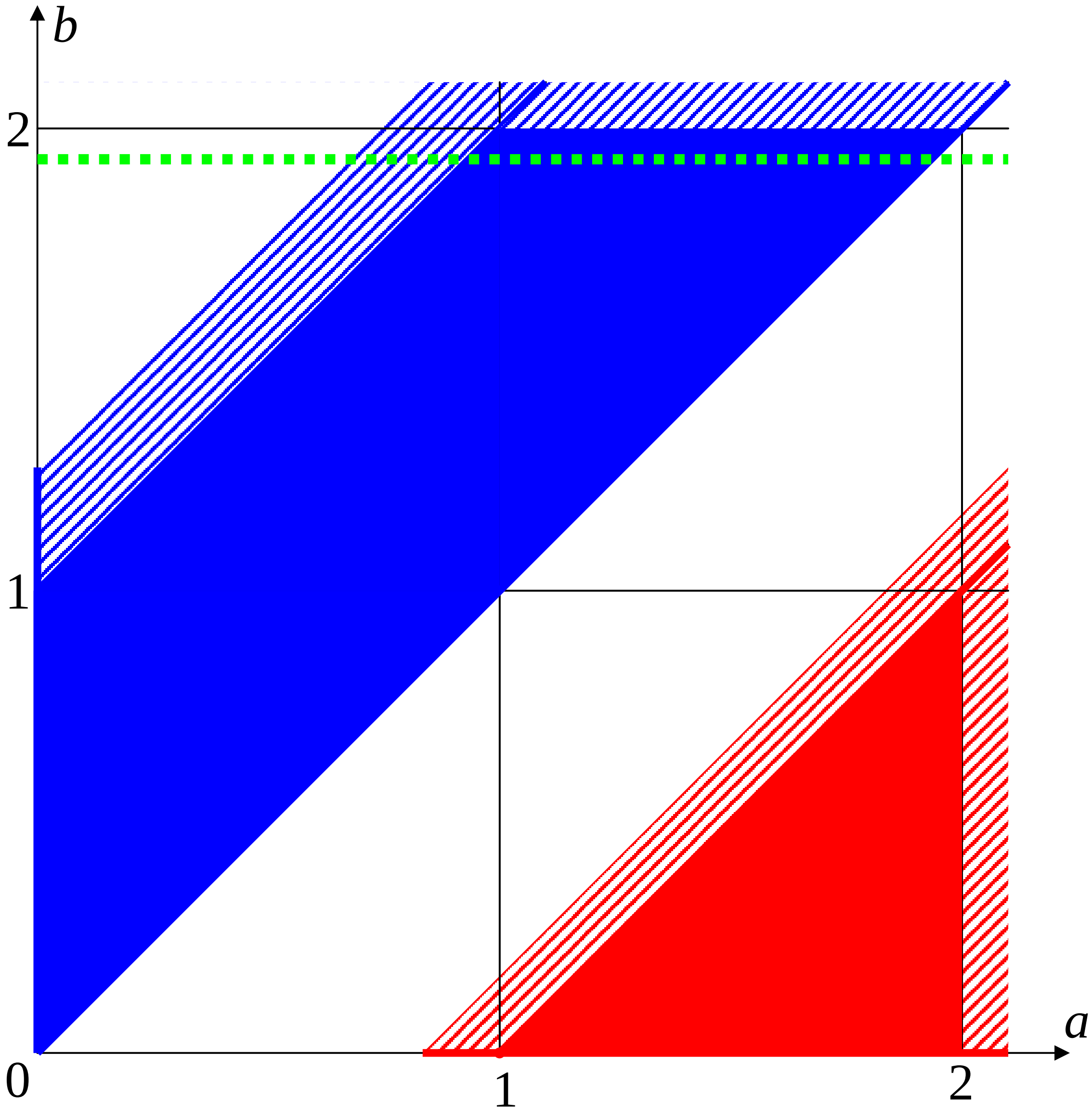
Example



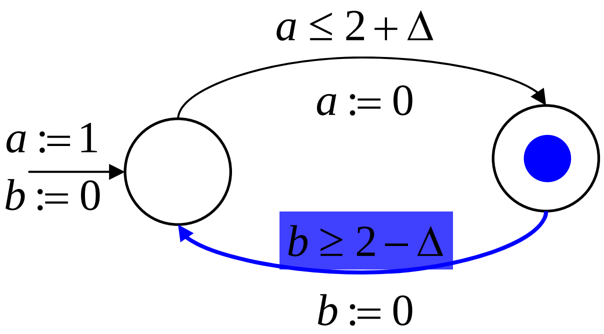
Enlarged Semantics



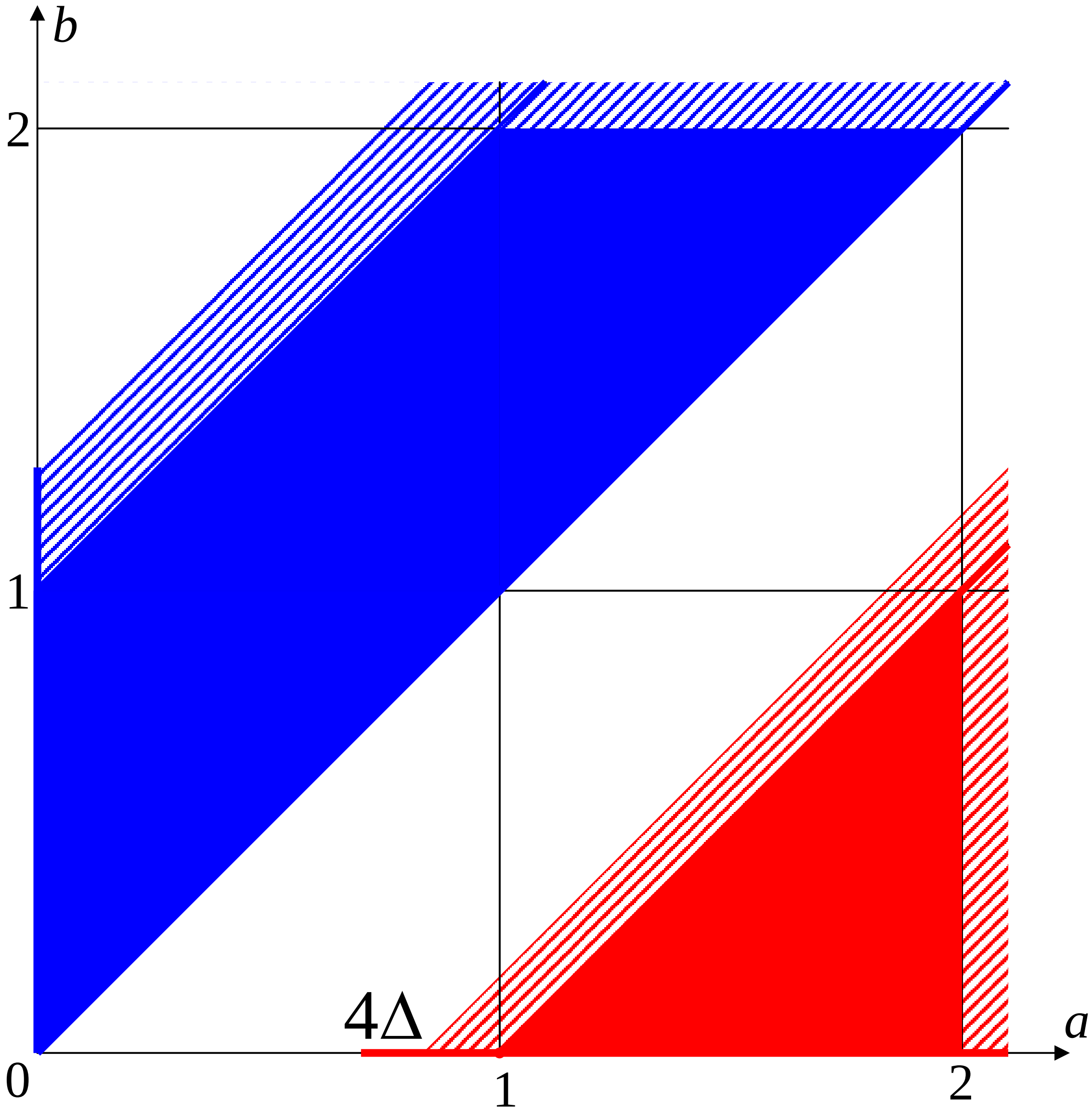
Example



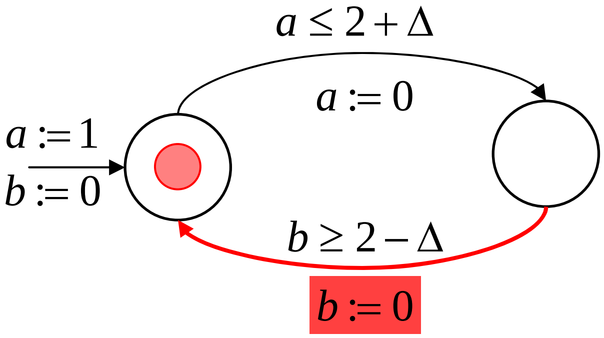
Enlarged Semantics



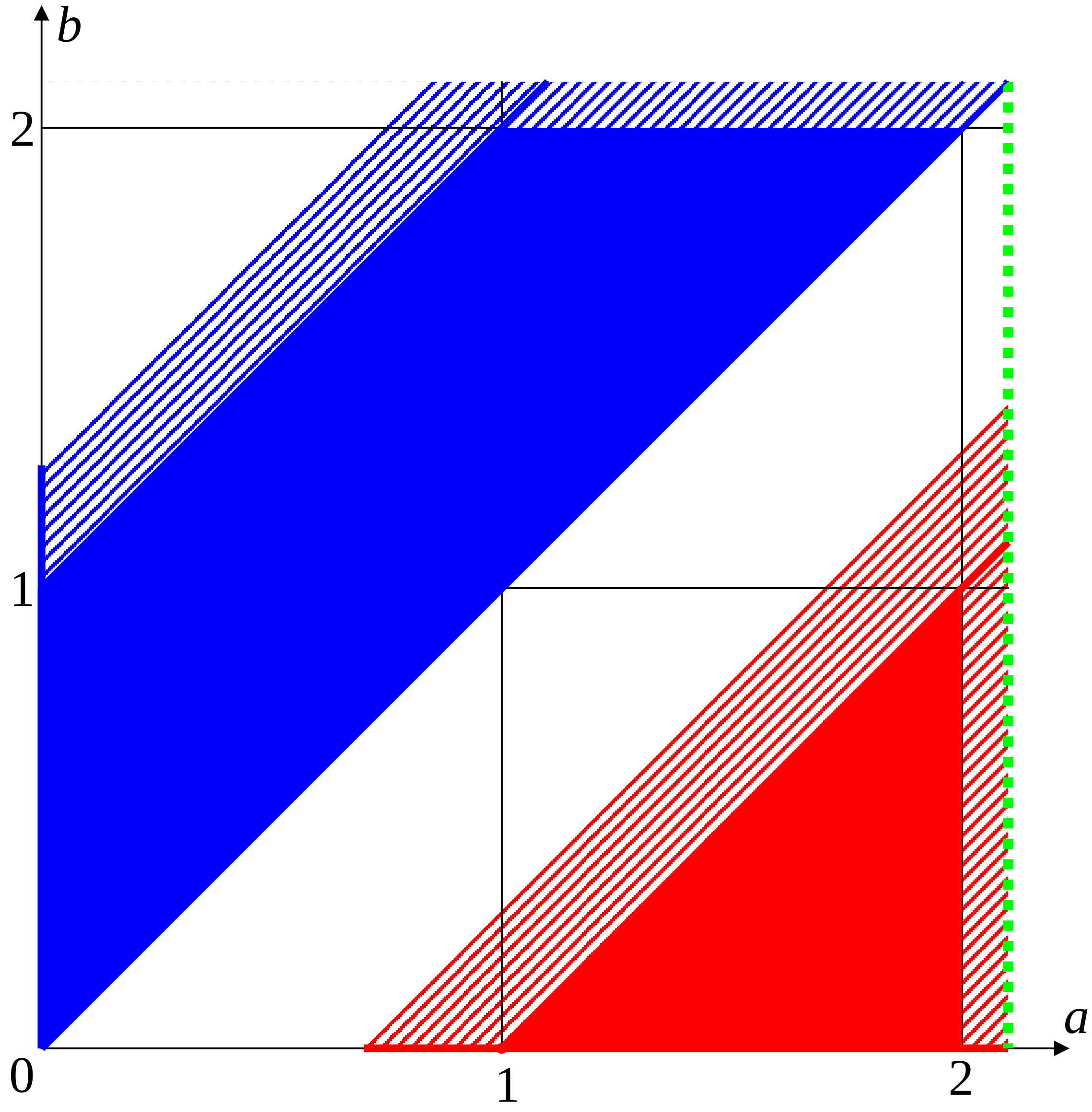
Example



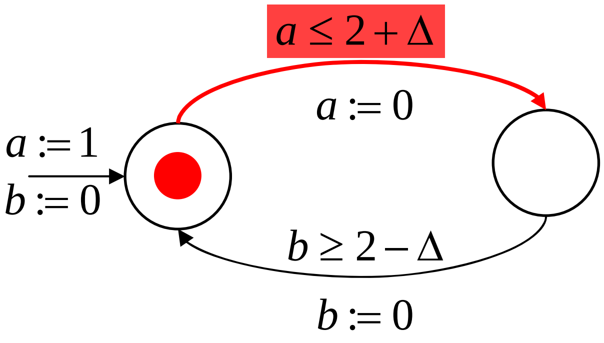
Enlarged Semantics



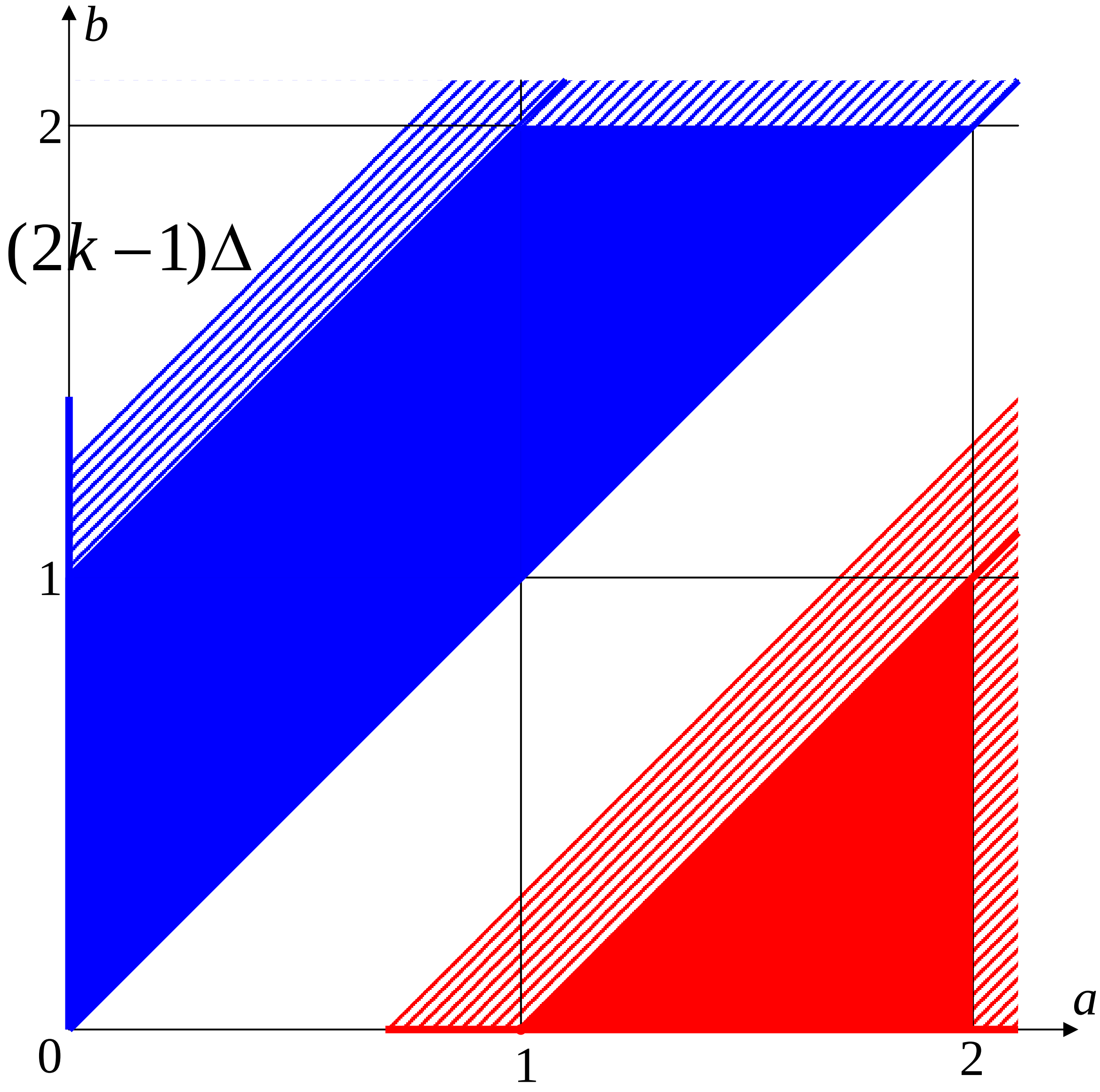
Example



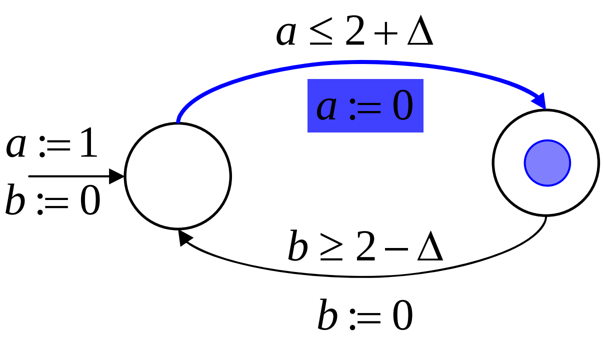
Enlarged Semantics



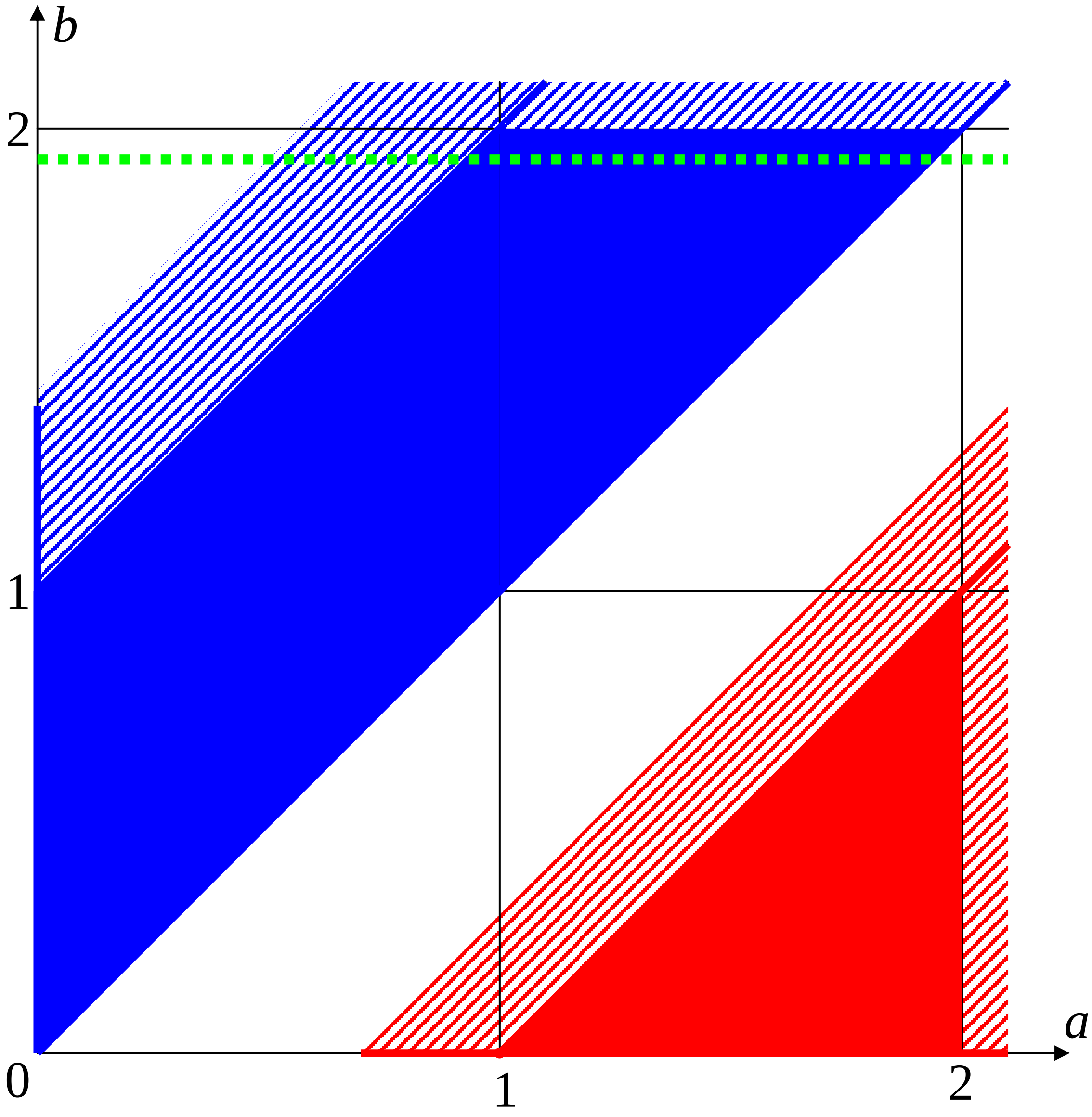
Example



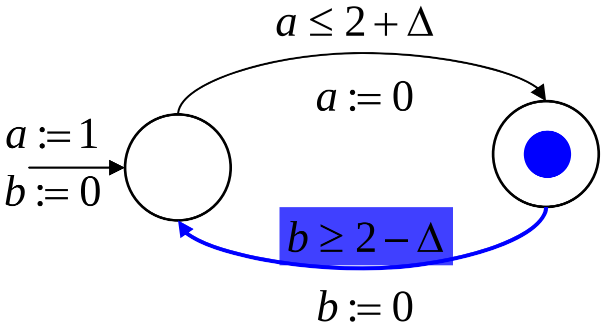
Enlarged Semantics



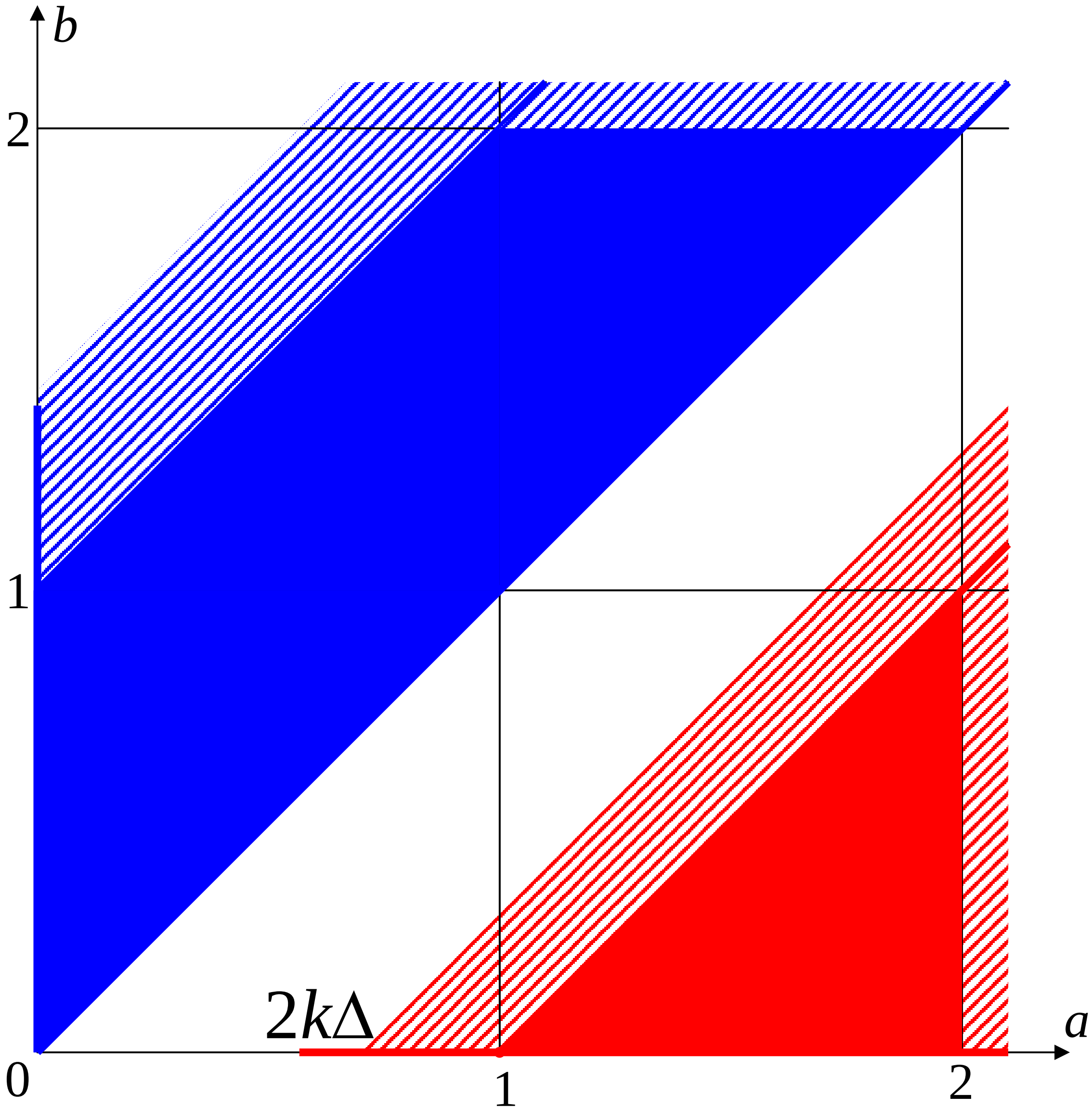
Example



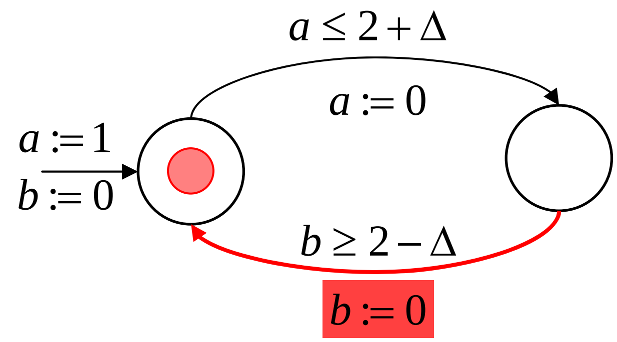
Enlarged Semantics



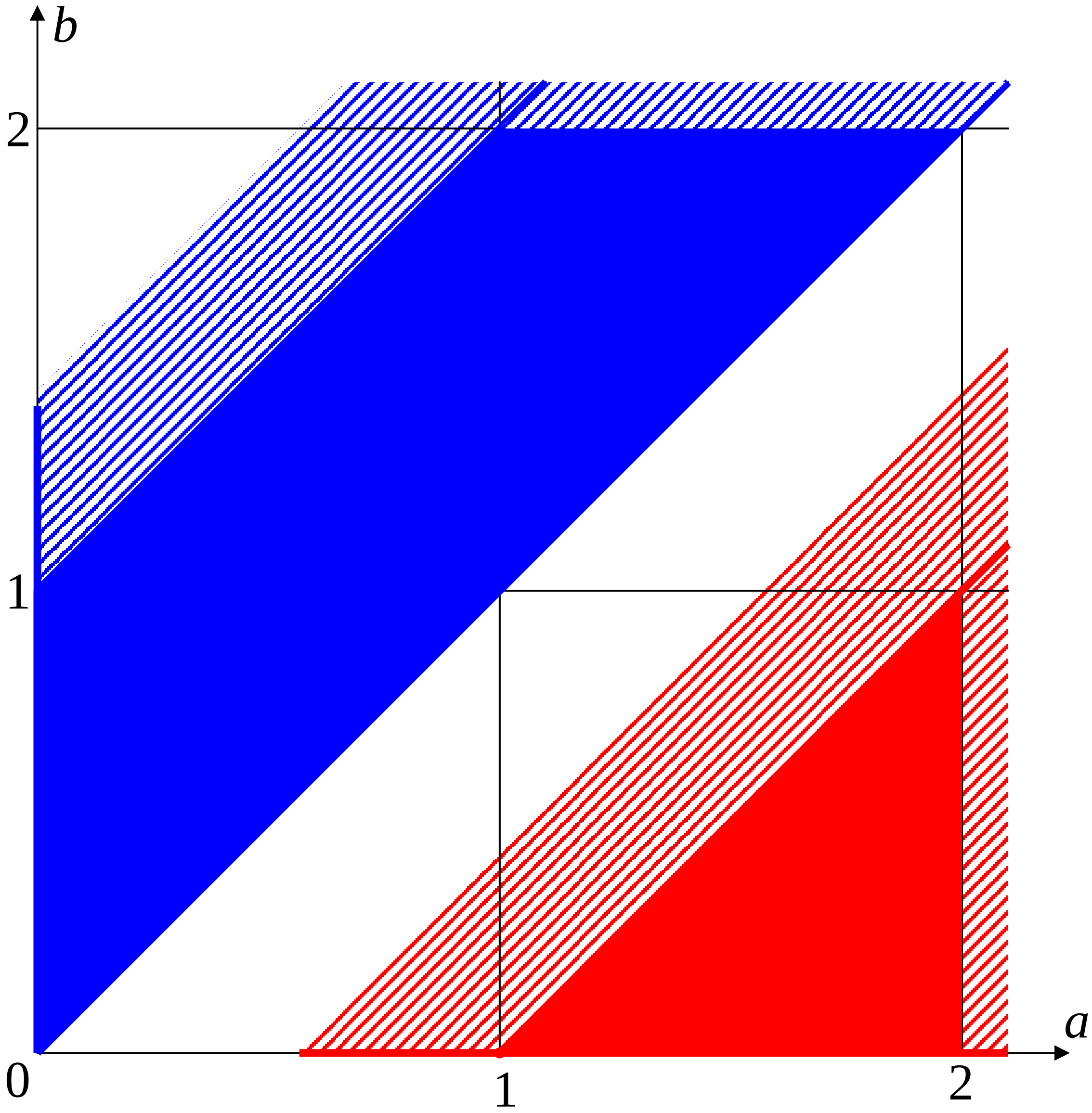
Example



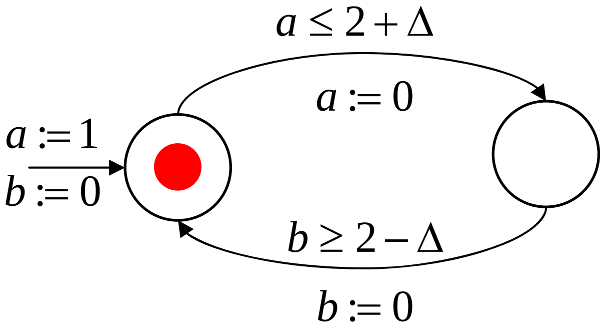
Enlarged Semantics



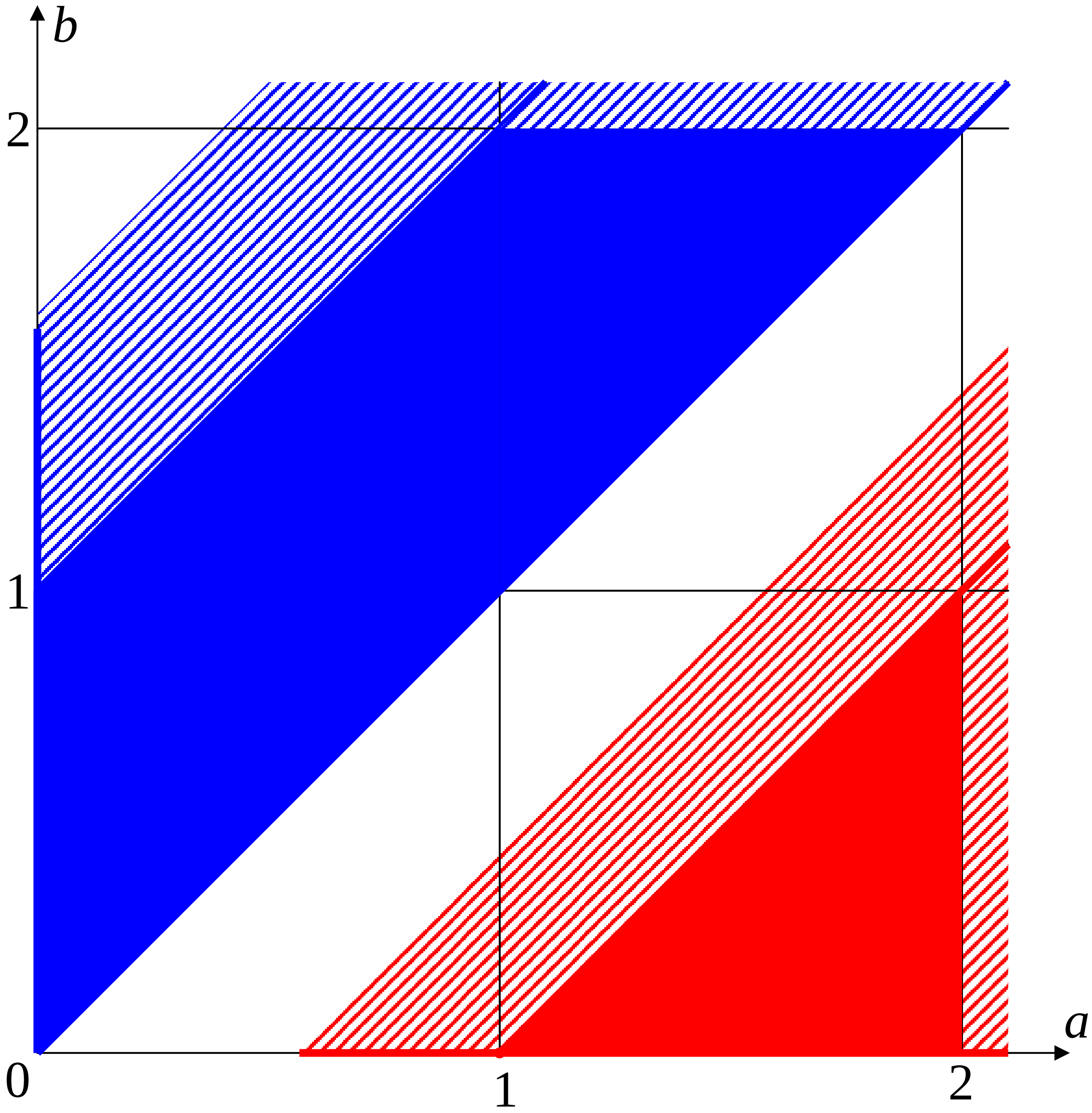
Example



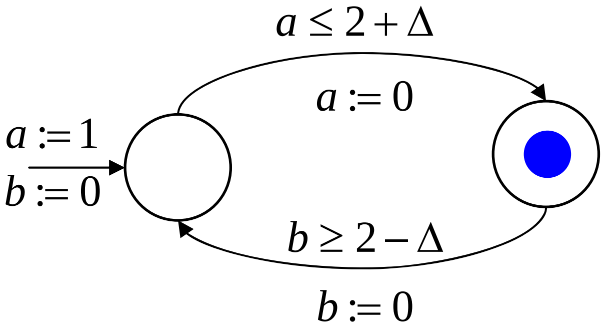
Enlarged Semantics



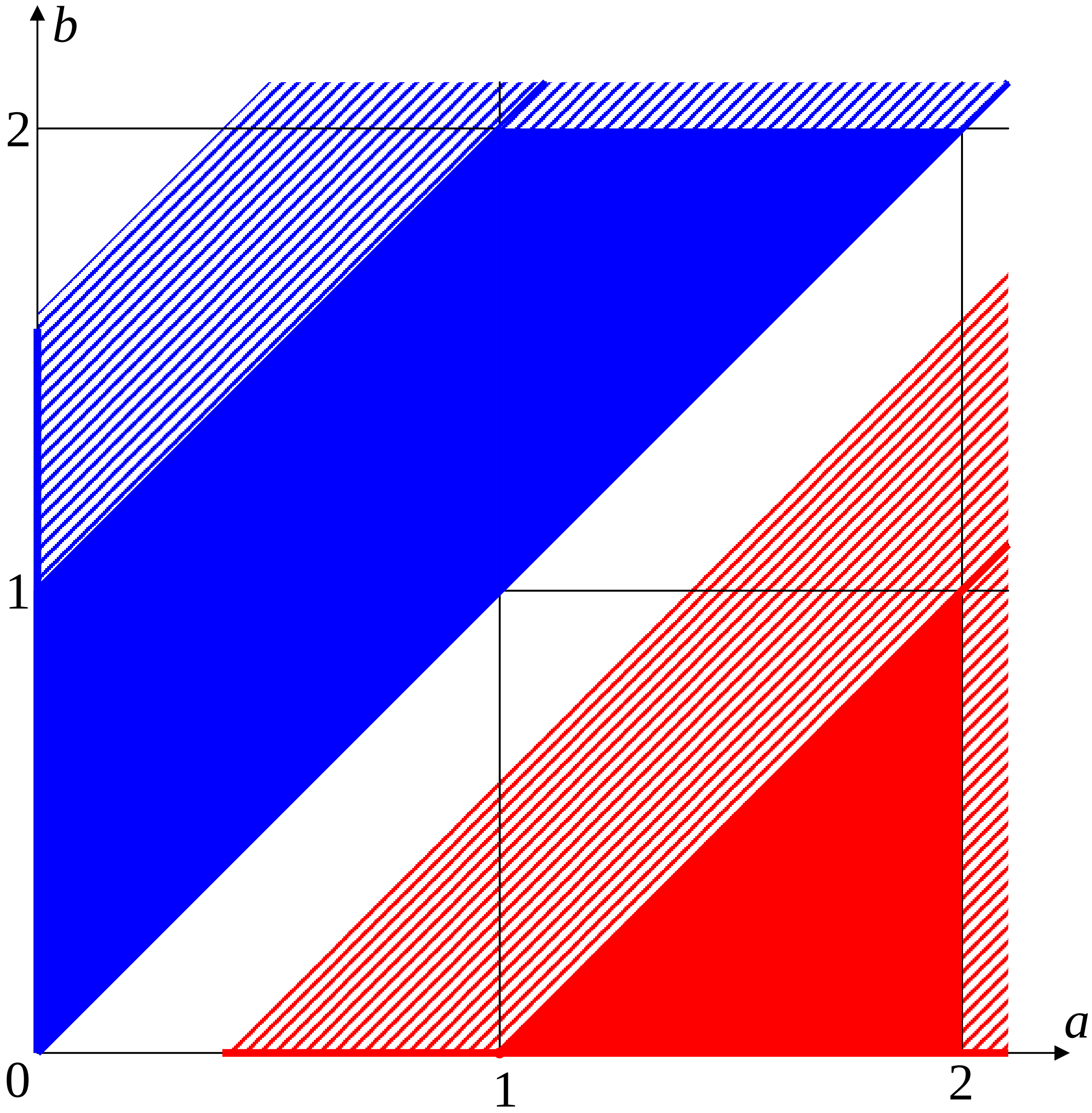
Example



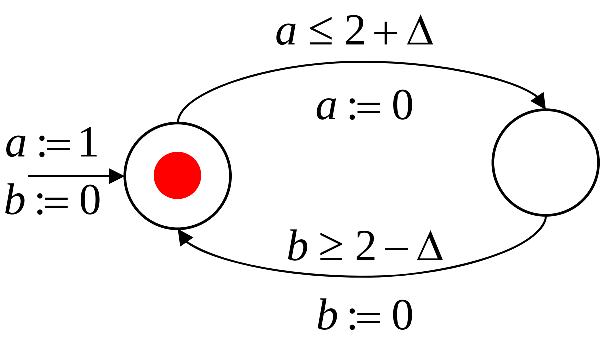
Enlarged Semantics



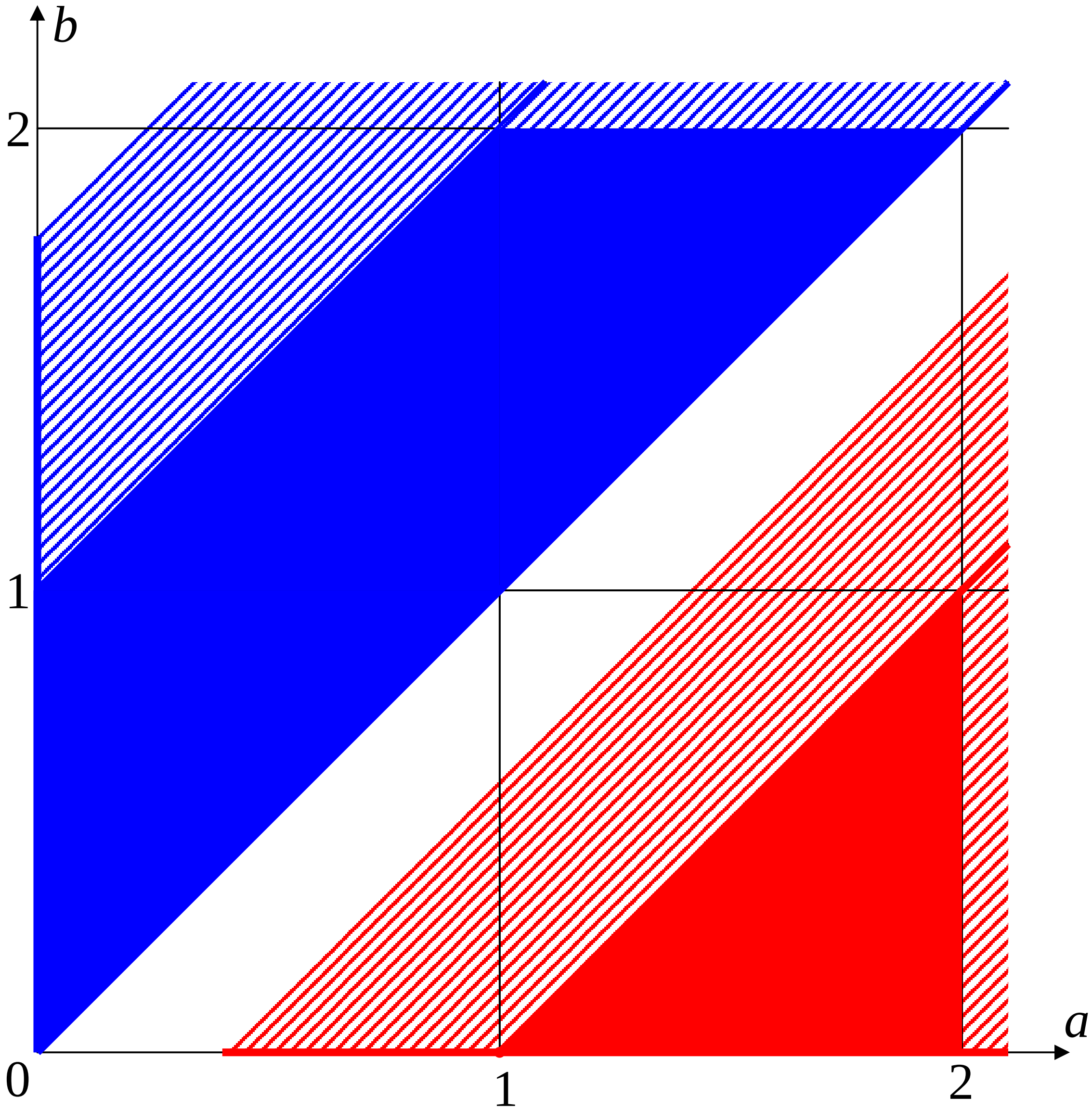
Example



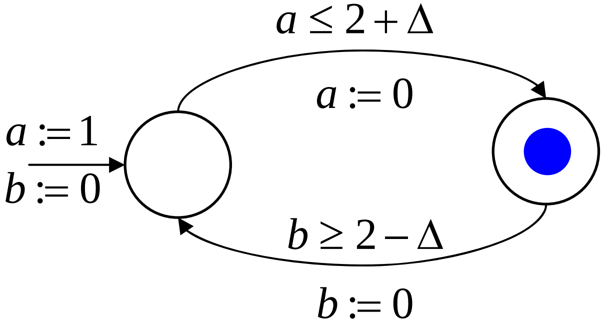
Enlarged Semantics



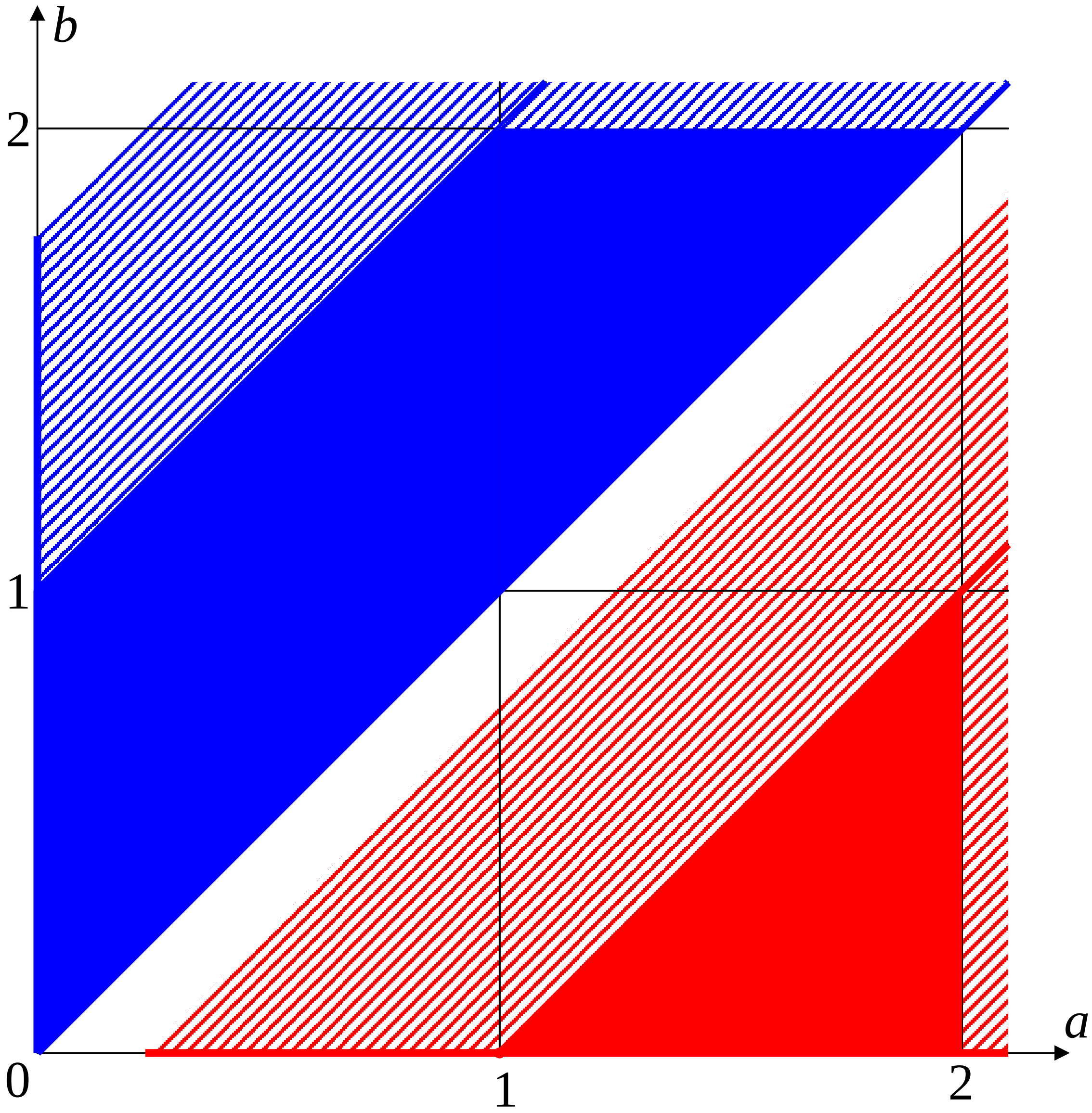
Example



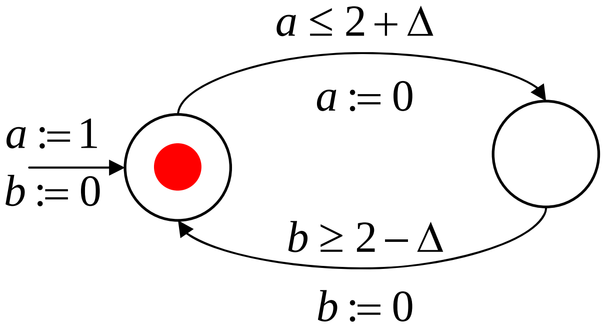
Enlarged Semantics



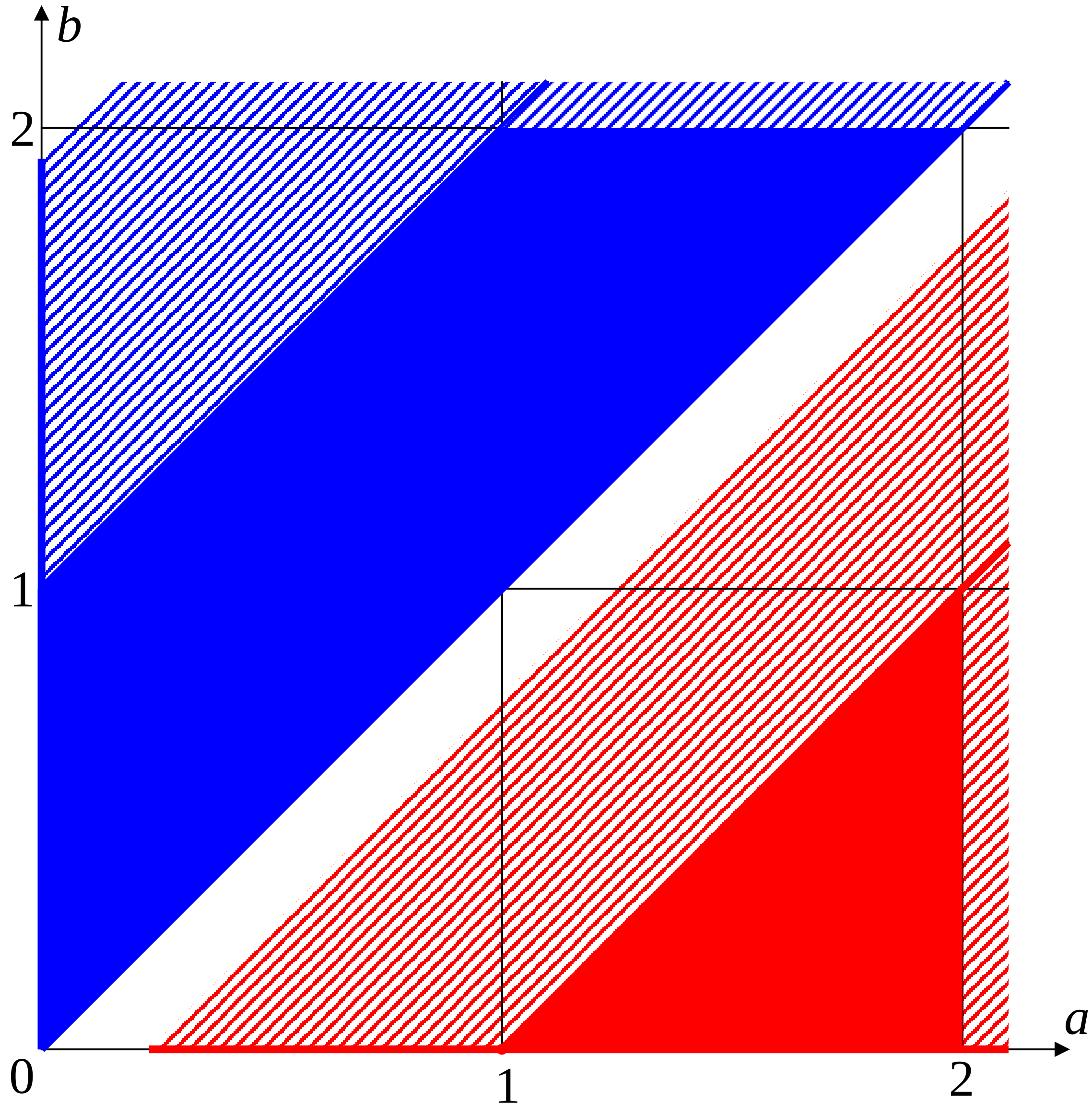
Example



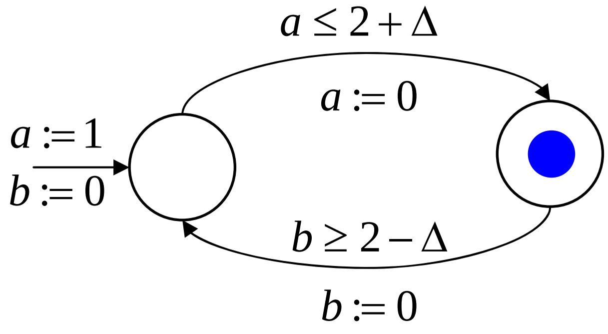
Enlarged Semantics



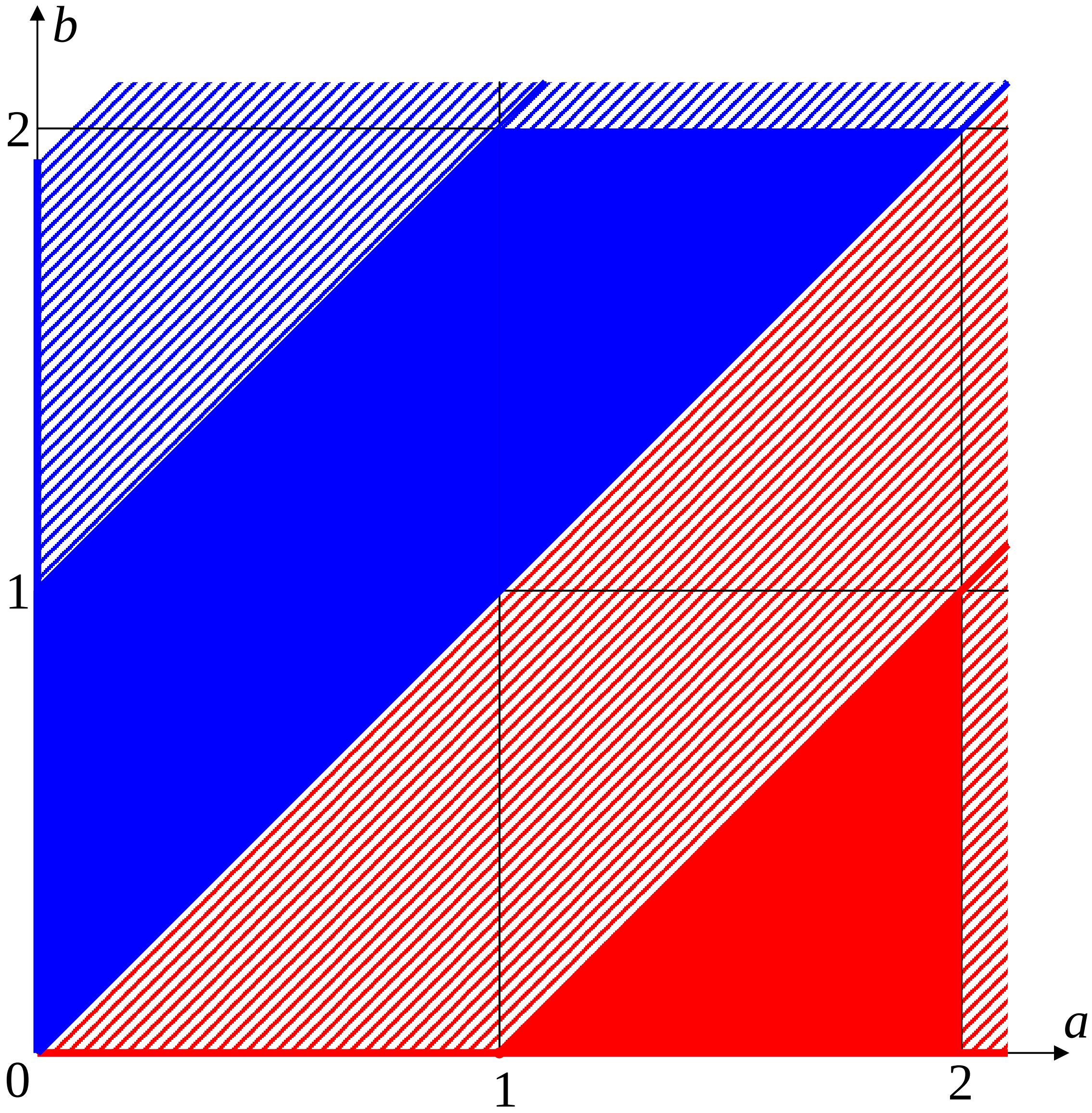
Example



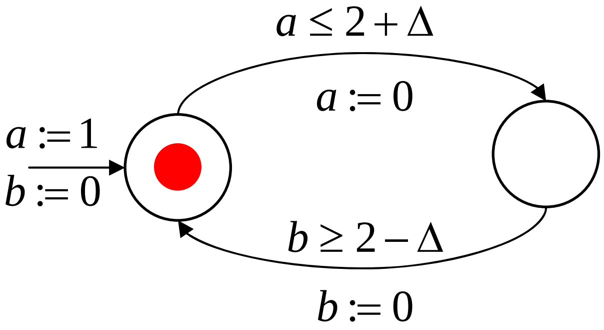
Enlarged Semantics



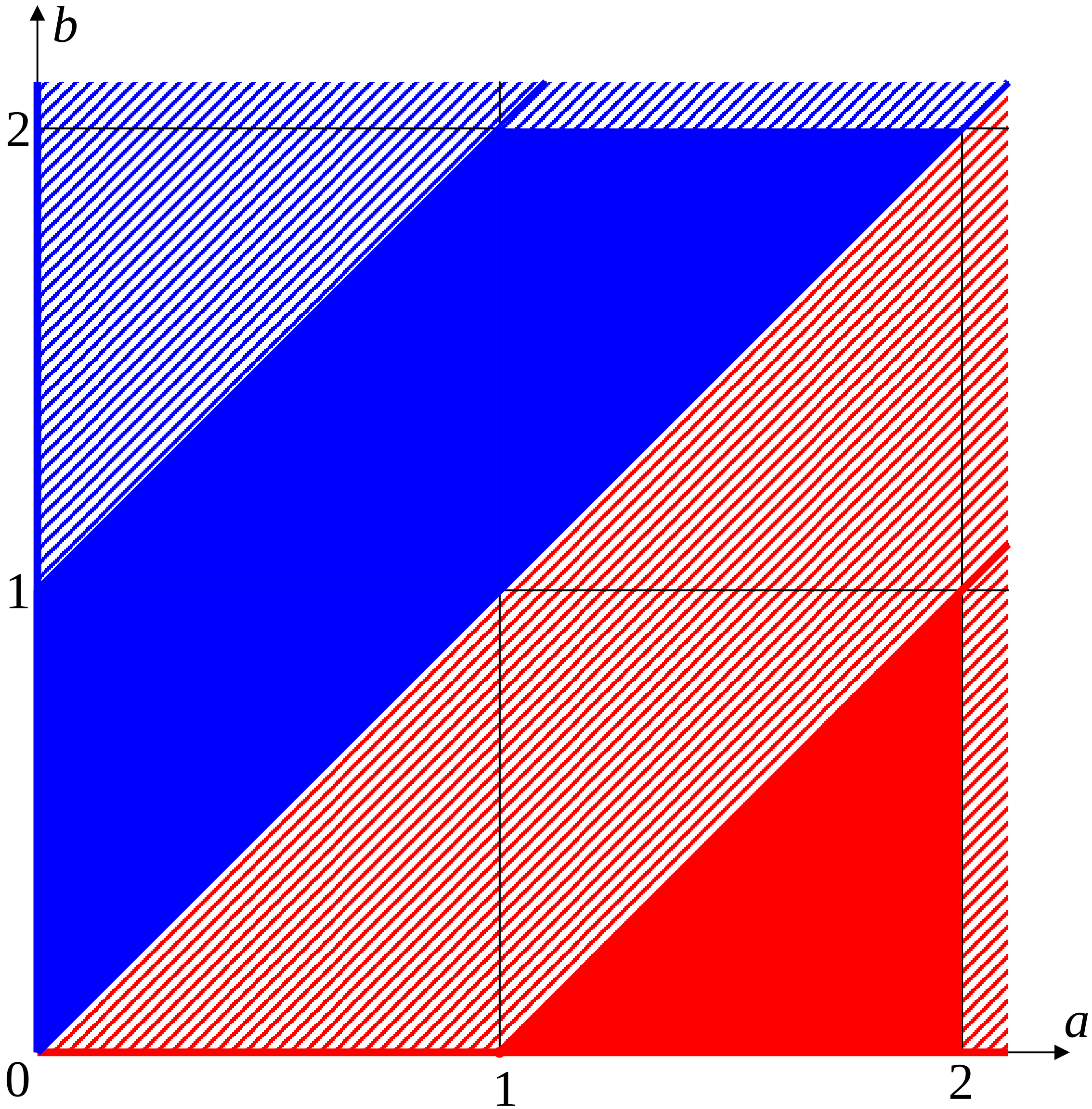
Example



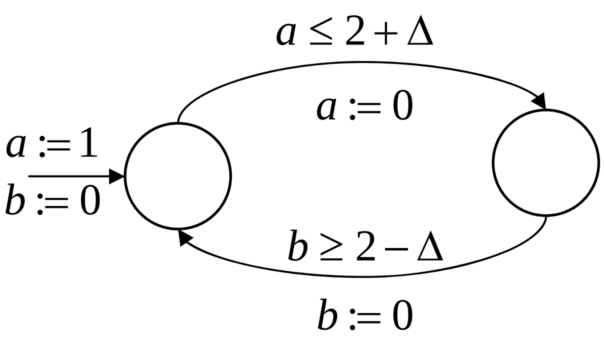
Enlarged Semantics



Example

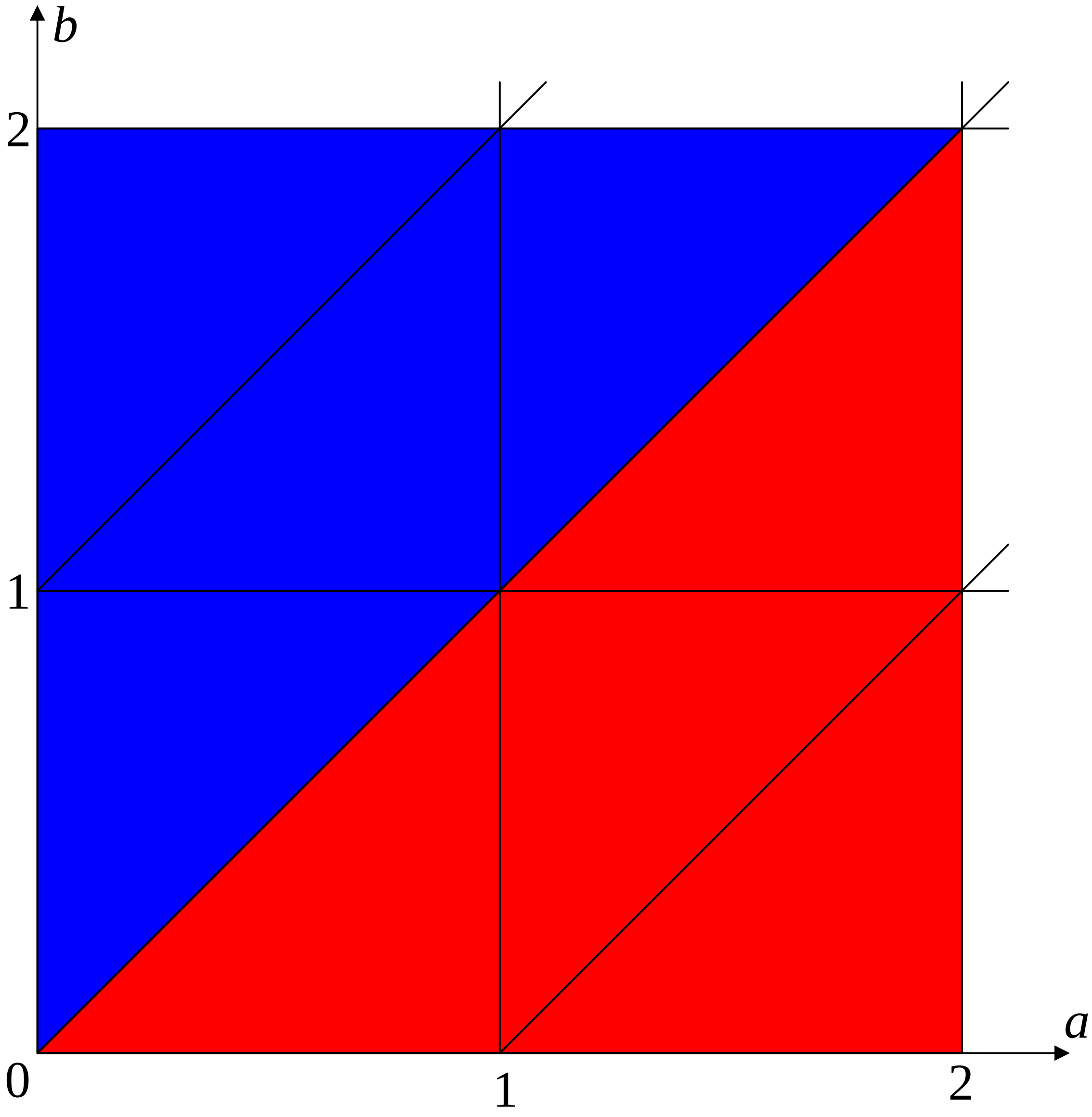


Enlarged Semantics



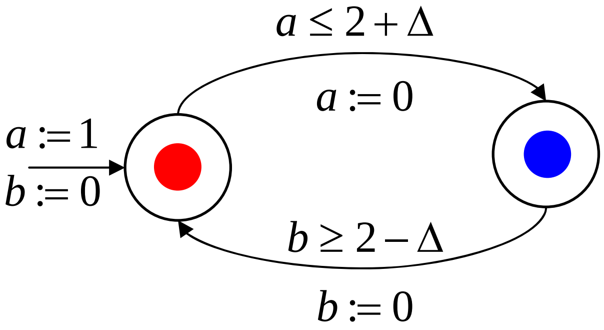
$\text{Reach}([A]_\Delta)$

Example



Enlarged Semantics

When $\Delta \to 0$

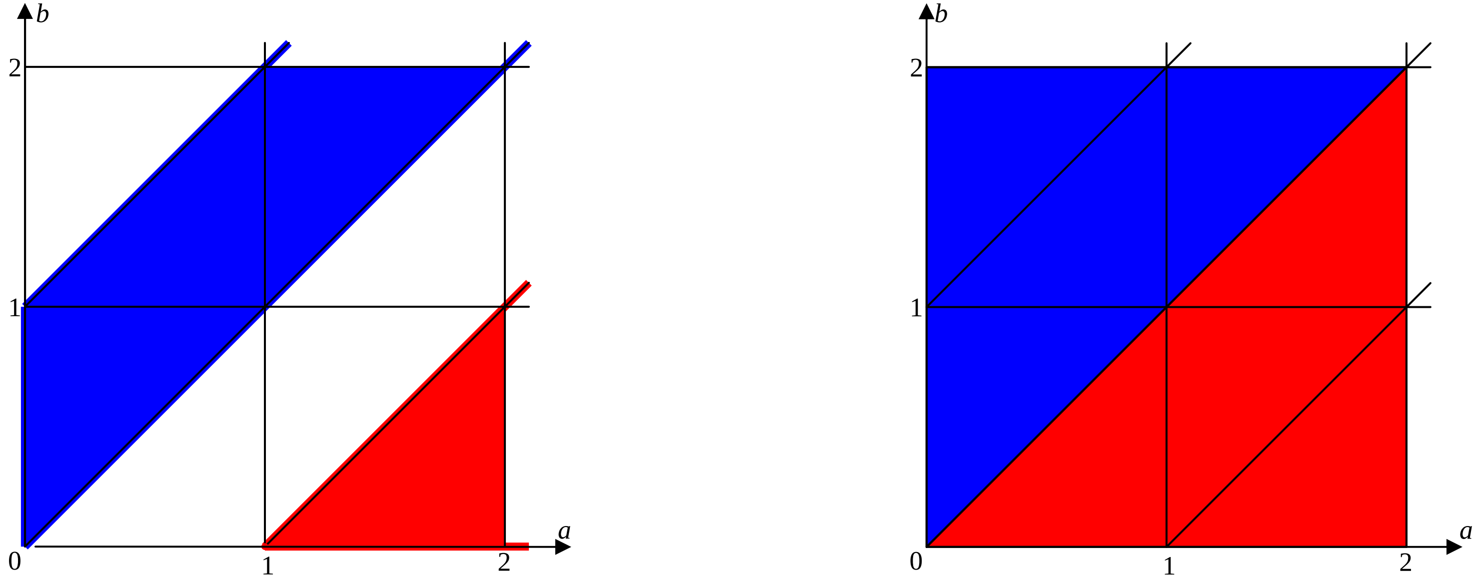


$$\bigcap_{\Delta>0} \mathrm{Reach}([A]_\Delta)$$

Example

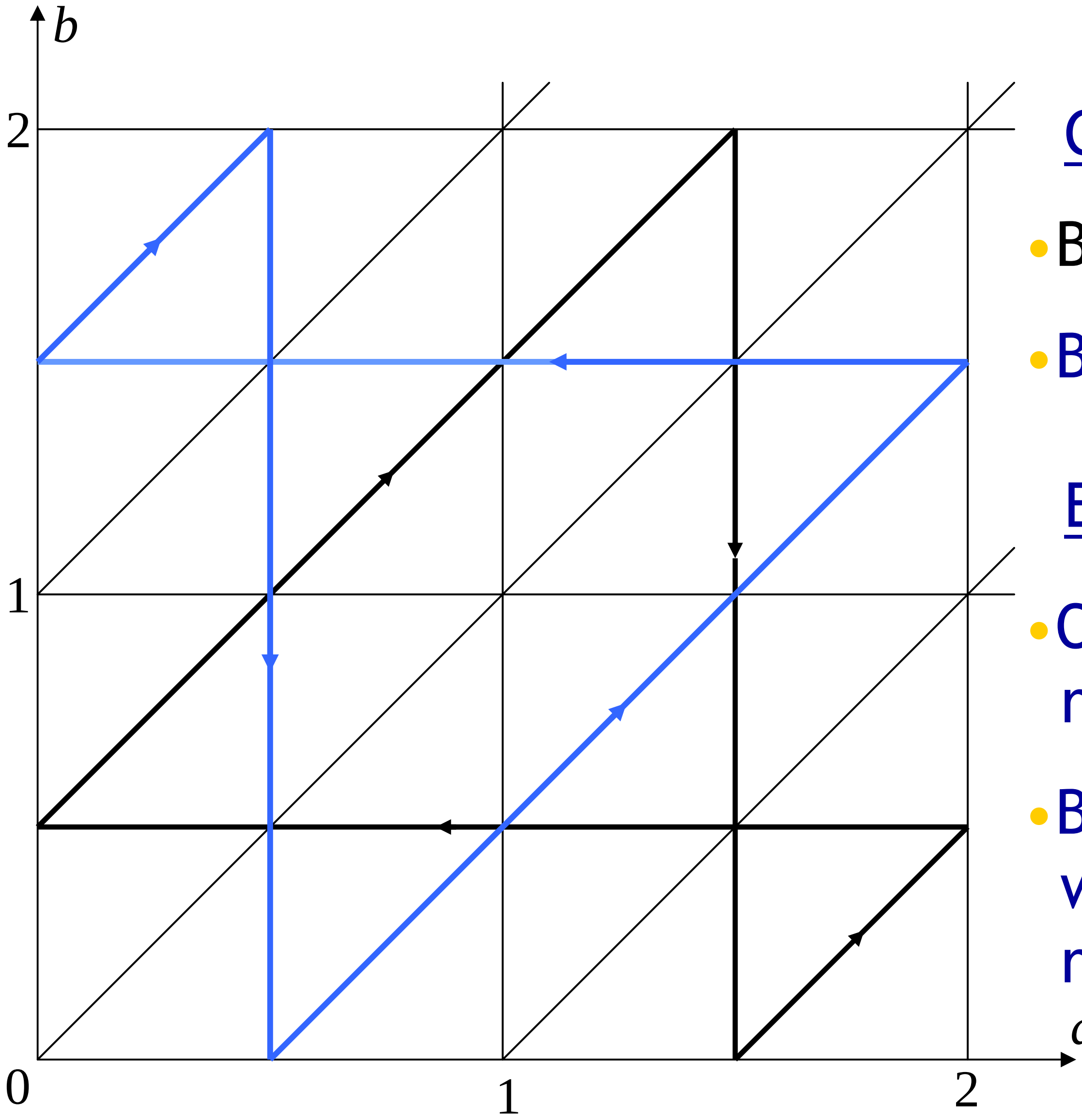
Classical Semantics vs. Enlarged Semantics

$\Delta = 0$ $\Delta \to 0$



$\text{Reach}([A]) \neq \bigcap_{\Delta>0} \text{Reach}([A]_\Delta)$

Example



Classical semantics [A]

- Black cycles are reachable
- Blue cycles are not !

Enlarged semantics $[A]_{\Delta}$

- One blue cycle is reachable
- By repeating this cycle with Δ>0, the entire regions are reachable !

Cycles in Timed Automata

Algortihm [Pur98] is based on this observation:

- It just adds the cycles to reachable states
- Until no more cycle is accessible

Hence, the *implementability problem*

Given a timed automaton A, determine whether there exists Δ>0 such that $\mathrm{Reach}([A]_\Delta) \cap \mathrm{Bad} = \emptyset$

is decidable ! (and PSPACE-complete)

Open questions

- Maximize Δ such that $\text{Reach}([A]_\Delta) \cap \text{Bad} = \varnothing$
- Decide whether there exists Δ such that

$$\text{UntimedLang}([A]_\Delta) = \text{UntimedLang}([A])$$

- Find a practical algorithm for $\bigcap_{\Delta>0} \text{Reach}([A]_\Delta)$
- And many others…

References

- [DDR04] M. De Wulf, L. Doyen, J.-F. Raskin. Almost ASAP Semantics: From Timed Model to Timed Implementation. LNCS 2993, HSCC 2004.

- [Pur98] A. Puri. Dynamical Properties of Timed Automata. FTRTFT 1998.

Model-based development

- Make a model of the environment:
  Env
- Make clear the control objective:
  Bad
- Make a model of the control strategy:
  ControllerModel
- Verify:
  Does Env || ControllerModel avoid Bad ?
- Good, but after ?