# Contents

# 1 Introduction

# 2 Basic Definitions

## 2.1 $\lambda$-calculus $\lambda 2$

In the following let $\mathcal{V}_T = \{\alpha, a, \beta, b, \dots\}$ be a countable set (of type-variables) and $\mathcal{V}_V = \{x_1, x_2, \dots\}$ be a countable set (of value-variables).

**Definition 1.** The set of all $\lambda 2$ types over $\mathcal{V}_T$, denoted by $\mathrm{T}_{\lambda 2}$, is the smallest set $\mathrm{T}$ satisfying the following conditions:

- $\mathcal{V}_T \subseteq \mathrm{T}$,
- if $t_1, t_2 \in \mathrm{T}$ then $(t_1 \to t_2) \in \mathrm{T}$, and
- if $t \in \mathrm{T}$ and $\alpha \in \mathcal{V}_T$ then $\forall \alpha . t \in \mathrm{T}$.

The set of all $\lambda 2$ terms over $\mathcal{V}_T$ and $\mathcal{V}_V$, denoted by $\Lambda_{\mathrm{T}_{\lambda 2}}$, is the smallest set $\Lambda_\mathrm{T}$ satisfying the following conditions:

- $\mathcal{V}_V \subseteq \Lambda_\mathrm{T}$,
- if $e_1, e_2 \in \Lambda_\mathrm{T}$ then $e_1 e_2 \in \Lambda_\mathrm{T}$,
- if $x \in \mathcal{V}_V$, $t \in \mathrm{T}_{\lambda 2}$, and $e \in \Lambda_\mathrm{T}$ then $\lambda x : t.e \in \Lambda_\mathrm{T}$,
- if $\alpha \in \mathcal{V}_T$ and $e \in \Lambda_\mathrm{T}$ then $\Lambda \alpha . e \in \Lambda_\mathrm{T}$, and
- if $e \in \Lambda_\mathrm{T}$ and $t \in \mathrm{T}_{\lambda 2}$ then $e\, t \in \Lambda_\mathrm{T}$.

**Definition 2.** Let $e \in \Lambda_{\mathrm{T}_{\lambda 2}}$. The free variables of $e$, denoted by $\mathrm{FV}(e)$, are defined inductively as follows:

$$\mathrm{FV}(e) = \begin{cases} \{x\} & \text{if } e = x \\ \mathrm{FV}(e_1) \cup \mathrm{FV}(e_2) & \text{if } e = e_1 e_2 \\ \mathrm{FV}(e') \setminus \{x\} & \text{if } e = \lambda x : t.e' \\ \mathrm{FV}(e') & \text{if } e = \Lambda \alpha . e' \\ \mathrm{FV}(e') & \text{if } e = e'\, t \end{cases}$$

Or is this definition better?

**Definition 3.** Let $e \in \Lambda_{\mathrm{T}_{\lambda 2}}$. The free variables of $e$, denoted by $\mathrm{FV}(e)$, are defined inductively as follows:

$$\begin{aligned} \mathrm{FV}(y) &= \{x\} \\ \mathrm{FV}(e_1 e_2) &= \mathrm{FV}(e_1) \cup \mathrm{FV}(e_2) \\ \mathrm{FV}(\lambda x : t.e') &= \mathrm{FV}(e') \setminus \{x\} \\ \mathrm{FV}(\Lambda \alpha . e') &= \mathrm{FV}(e') \\ \mathrm{FV}(e'\, t) &= \mathrm{FV}(e') \end{aligned}$$

**Definition 4.** Let $\mathcal{V} = \{x_1, \dots, x_n\}$ be a finite subset of $\mathcal{V}_T$ and $t_1, \dots, t_n \in \Lambda_{\mathrm{T}_{\lambda 2}}$. A $\boldsymbol{\lambda 2}$-basis $\Gamma = \{(x_1 : t_1), \dots, (x_n : t_n)\}$ is a mapping from $\mathcal{V}$ to $\mathrm{T}_{\lambda 2}$. If the kind of basis is clear from the context we abbreviate $\boldsymbol{\lambda 2}$-basis to basis.

The free variables of a basis $\Gamma$, denoted by $\mathrm{FV}(\Gamma)$, are $\bigcup\{\mathrm{FV}(t) \mid (x : t) \in \Gamma\}$.

For a basis $\Gamma$, $x \in \mathcal{V}_V \setminus \mathrm{dom}(\Gamma)$, and $t \in \mathrm{T}_{\lambda 2}$ we will abbreviate $\Gamma \cup \{(x : t)\}$ to $\Gamma, x : t$.

**Definition 5.** Let $e$ be in $\Lambda_{\mathrm{T}_{\lambda 2}}$, $t$ in $\mathrm{T}_{\lambda 2}$, and $\Gamma$ be a basis. A statement $e : t$ is derivable from $\Gamma$, denoted by $\Gamma \vdash e : t$, if $e : t$ can be produced using the following rules.

| | | |
|---|---|---|
| (Axiom) | $\Gamma, x : t \vdash x : t$ | |
| ($\lambda$-Introduction) | $\dfrac{\Gamma, x : t_1 \vdash e : t_2}{\Gamma \vdash \lambda x : t_1.e : t_1 \to t_2}$ | |
| ($\lambda$-Elimination) | $\dfrac{\Gamma \vdash e_1 : t_1 \to t_2 \quad \Gamma \vdash e_2 : t_1}{\Gamma \vdash e_1 e_2 : t_2}$ | |
| ($\forall$-Introduction) | $\dfrac{\Gamma \vdash e : t}{\Gamma \vdash \Lambda\alpha.e : \forall\alpha.t}$ | $\alpha \notin \mathrm{FV}(\Gamma)$ |
| ($\forall$-Elimination) | $\dfrac{\Gamma \vdash e : \forall\alpha.t}{\Gamma \vdash e\,t' : t\,[\alpha := t']}$ | $t' \in \mathrm{T}_{\lambda 2}$ |

**Definition 6.** The inhabitation problem for $\lambda 2$, denoted by **INHAB**, is defined as follows. Given a $\lambda 2$ type $t$.

$$\text{Is there a } \lambda 2 \text{ term } M \text{ such that } \emptyset \vdash M : t?$$

But we can rephrase this problem so that it becomes more general: Given a basis $\Gamma$ and a $\lambda 2$ type $t$.

$$\text{Is there a } \lambda 2 \text{ term } M \text{ such that } \Gamma \vdash M : t?$$

Obviously the second version is a special case of the first one. For the other direction consider a basis $\Gamma = \{(x_1 : t_1), \dots, (x_n : t_n)\}$ and a $\lambda 2$ type $t$. Clearly, for every term $M$, $\Gamma \vdash M : t$ holds iff $\emptyset \vdash \lambda x_1 : t_1. \dots \lambda x_n : t_n.M : t_1 \to \dots \to t_n \to t$.

## 2.2 first-order logic

**Definition 7.** A ranked set is a tuple $(\Sigma, rk)$, where $\Sigma$ is a countable set and $rk\colon \Sigma \to \mathbb{N}$ is a function that maps every symbol from $\Sigma$ to a natural number (its rank).

If the function $rk$ is understood we will just write $\Sigma$ instead of $(\Sigma, rk)$. The set of all elements in $\Sigma$ with a certain rank $k$, denoted by $\Sigma^{(k)}$, is defined as $\Sigma^{(k)} := rk^{-1}(k)$.

For the remainder of this subsection let $\mathcal{V} = \{y_1, y_2, \dots\}$ be a countable set (of variables), $\mathcal{F}$ a ranked set (of function symbols), and $\mathcal{P}$ a ranked set (of predicate symbols).

**Definition 8.** The set of terms over $\mathcal{V}$ and $\mathcal{F}$, denoted by $\mathcal{T}_{(\mathcal{V},\mathcal{F})}$, is the smallest set $\mathcal{T}$ satisfying the following conditions:

- $\mathcal{V} \subseteq \mathcal{T}$, and
- for every $k \in \mathbb{N}$ if $f \in \mathcal{F}^{(k)}$ and $t_1, t_2, \ldots, t_k \in \mathcal{T}$ then $f(t_1, t_2, \ldots, t_k) \in \mathcal{T}$.

The set of first-order formulas over $\mathcal{V}$, $\mathcal{F}$, and $\mathcal{P}$, denoted by $\mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$, is the smallest set $\mathcal{L}$ satisfying the following conditions:

- for every $k \in \mathbb{N}$ if $P \in \mathcal{P}^{(k)}$ and $t_1, t_2, \ldots, t_k \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$ then $P(t_1, t_2, \ldots, t_k) \in \mathcal{L}$.
- If $\varphi, \psi \in \mathcal{L}$ then $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\neg\varphi \in \mathcal{L}$, and
- if $y \in \mathcal{V}$ and $\varphi \in \mathcal{L}$ then $\exists y.\varphi$, $\forall y.\varphi \in \mathcal{L}$.

We introduce an additional binary operation $\rightarrow$ on formulas, where for some $\varphi$, $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$ the formula $(\varphi \rightarrow \psi)$ is defined as $(\neg\varphi \vee \psi)$. For nullary relation symbols $P$ we will abbreviate $P()$ to $P$. If a formula $\varphi$ is of the form $Qx.(\psi)$ (where $Q \in \{\exists, \forall\}$, $x \in \mathcal{V}$, and $(\psi) \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$) we often drop the dot and write $Qx(\psi)$ instead. If a formula $\varphi$ has multiple variables binded by the same quantifier (i.e. $\varphi = Qx_1.Qx_2.\ldots Qx_n.\psi$ for $Q \in \{\exists, \forall\}$, some $n \in \mathbb{N}$, $x_1, x_2, \ldots, x_n \in \mathcal{V}$, and $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$) we abbreviate $\varphi$ to $Qx_1x_2 \ldots x_n.\psi$ or to $Q\vec{x}.\psi$ where $\vec{x} = (x_1, x_2, \ldots, x_n)^\top$.

**Definition 9.** The variables of a term $t \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$, denoted by $\mathrm{V}(t)$, are defined by:

$$\mathrm{V}(t) = \begin{cases} \{y\} & \text{if } t = y \\ \mathrm{V}(t_1) \cup \mathrm{V}(t_2) \cup \cdots \cup \mathrm{V}(t_k) & \text{if } t = f(t_1, t_2, \ldots, t_k) \end{cases}$$

The free variables of a formula $\varphi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$, denoted by $\mathrm{FV}(\varphi)$, are defined as follows:

$$\mathrm{FV}(\varphi) = \begin{cases} \mathrm{V}(t_1) \cup \mathrm{V}(t_2) \cup \cdots \cup \mathrm{V}(t_k) & \text{if } \varphi = P(t_1, t_2, \ldots, t_k) \\ \mathrm{FV}(\psi) & \text{if } \varphi = \neg\psi \\ \mathrm{FV}(\varphi_1) \cup \mathrm{FV}(\varphi_2) & \text{if } \varphi = (\varphi_1 \wedge \varphi_2) \text{ or } \varphi = (\varphi_1 \vee \varphi_2) \\ \mathrm{FV}(\psi) \setminus \{y\} & \text{if } \varphi = \forall y.\psi \text{ or } \varphi = \exists y.\psi \end{cases}$$

**Definition 10.** Let $y$ be in $\mathcal{V}$ and $t, t' \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$. The substitution of $y$ by $t'$ in $t$, denoted by $t\left[y := t'\right]$, is defined as follows:

$$t\left[y := t'\right] = \begin{cases} t' & \text{if } t = y \\ z & \text{if } t = z \text{ and } z \neq y \\ f(t_1\left[y := t'\right], \ldots, t_k\left[y := t'\right]) & \text{if } t = f(t_1, \ldots, t_k) \end{cases}$$

Now we can lift this definition to formulas, let $\varphi$ be in $\mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$. The substitution of $y$ by $t'$ in $\varphi$, denoted by $\varphi\,[y := t']$, is defined as follows:

$$\varphi\,[y := t'] = \begin{cases} P(t_1\,[y := t'],\ldots,t_k\,[y := t']) & \text{if } \varphi = P(t_1,\ldots,t_k) \\ \neg(\psi\,[y := t']) & \text{if } \varphi = \neg\psi \\ \varphi_1\,[y := t'] \circ \varphi_2\,[y := t'] & \text{if } \varphi = (\varphi_1 \circ \varphi_2)\ ,\ \circ \in \{\wedge,\vee\} \\ \varphi & \text{if } \varphi = \forall y.\psi \text{ or } \varphi = \exists y.\psi \\ Qz.(\psi\,[y := t']) & \text{if } \varphi = Qz.\psi,\ Q \in \{\forall,\exists\} \text{ and } z \neq y \end{cases}$$

Now we come to the semantics of first-order formulas.

**Definition 11.** An interpretation $I$ over $\mathcal{V}$, $\mathcal{F}$, and $\mathcal{P}$ is a triple $(\Delta,\cdot^I,\omega)$, where

| | | |
|---|---|---|
| where | $\Delta$ | is a nonempty set (which we call domain), |
| | $\cdot^I$ | is a function such that |
| | | $f^I\colon \Delta^k \to \Delta$ is a function for every $k \in \mathbb{N}$, $f \in \mathcal{F}^{(k)}$ and |
| | | $P^I \subseteq \Delta^k$ is a relation for every $k \in \mathbb{N}$, $f \in \mathcal{P}^{(k)}$ |
| | $\omega$ | is a function from $\mathcal{V}$ to $\Delta$. |

Let $I = (\Delta,\cdot^I,\omega)$ be an interpretation, $y \in \mathcal{V}$, and $d \in \Delta$ the interpretation $I\,[y \mapsto d]$ is defined as $(\Delta,\cdot^I,\omega\,[y \mapsto d])$ where

$$(\omega\,[y \mapsto d])(z) = \begin{cases} d & \text{if } z = y \\ \omega(y) & \text{otherwise.} \end{cases}$$

**Definition 12.** Let $I = (\Delta,\cdot^I,\omega)$ be an interpretation and $t$ a term. The interpretation of $t$ under $I$, denoted by $t^I$, is defined as follows:

$$t^I = \begin{cases} \omega(y) & \text{if } t = y \\ f^I(t_1^I,\ldots,t_k^I) & \text{if } t = f(t_1,\ldots,t_k) \end{cases}$$

Let $\varphi$ be a formula. The interpretation of $\varphi$ under $I$, denoted by $\varphi^I$, is defined recursively as follows:

$$\varphi^I = \begin{cases} \top & \text{if } \varphi = P(t_1,\ldots,t_k) \text{ and } (t_1^I,\ldots,t_k^I) \in P^I \\ \bot & \text{if } \varphi = P(t_1,\ldots,t_k) \text{ and } (t_1^I,\ldots,t_k^I) \notin P^I \\ \text{not } \psi^I & \text{if } \varphi = \neg\psi \\ \varphi_1^I \text{ and } \varphi_2^I & \text{if } \varphi = (\varphi_1 \wedge \varphi_2) \\ \varphi_1^I \text{ or } \varphi_2^I & \text{if } \varphi = (\varphi_1 \vee \varphi_2) \\ \text{exists } d \in \Delta\ \psi^{I[y\mapsto d]} & \text{if } \varphi = \exists y.\psi \\ \text{forall } d \in \Delta\ \psi^{I[y\mapsto d]} & \text{if } \varphi = \forall y.\psi \end{cases}$$

The interpretation $I$ is a model of $\varphi$, denoted by $I \models \varphi$, if $\varphi^I = \top$.

When we define an interpretation $I$ and we have a nullary predicate symbol $P$ we write $P^I = \top$ instead of $P^I = \{()\}$ and $P^I = \bot$ for $P^I = \emptyset$ (this works because $P()^I = \top$ iff $() \in P^I$).

**Definition 13.** Let $\Gamma$ be a finite set of first-oder formulas.

We say that an interpretation $I$ is a model of $\Gamma$, denoted by $I \models \Gamma$, if $I \models \psi$ for every $\psi$ in $\Gamma$.

The formula $\varphi$ is a semantic consequence of $\Gamma$, denoted by $\Gamma \vdash \varphi$, if every model of $\Gamma$ is also a model of $\varphi$.

The free variables of $\Gamma$, denoted by $\mathrm{FV}(\Gamma)$, are $\bigcup\{\mathrm{FV}(\varphi) \mid \varphi \in \Gamma\}$.

### 2.3 two-counter automaton

We will use a version of two-counter automaton which only has two types of transitions. First it can increment a register and second it can decrement a register or jump if the register is already zero. Formally:

**Definition 14.** A deterministic two-counter automaton is a 4-tuple $M = (\mathcal{Q}, Q_0, Q_f, R)$,

| | | |
|---|---|---|
| where | $\mathcal{Q}$ | is a finite set (of states), |
| | $Q_0$ | is in $\mathcal{Q}$ (the initial state), |
| | $Q_f$ | is in $\mathcal{Q}$ (the final state), and |
| | $R$ | is a function from $\mathcal{Q} \setminus \{Q_f\}$ to $\mathcal{R}_{\mathcal{Q}}$, where $\mathcal{R}_{\mathcal{Q}} = \{+(i, Q') \mid i \in \{1,2\}, Q' \in \mathcal{Q}\} \cup \{-(i, Q_1, Q_2) \mid i \in \{1,2\}, Q_1, Q_2 \in \mathcal{Q}\}$ |

A configuration $C$ of our automaton is a triple $\langle Q, m, n\rangle$, where $Q \in \mathcal{Q}$ and $m, n \in \mathbb{N}$. Let $r$ be in $R(\mathcal{Q} \setminus \{Q_f\})$, then $\Rightarrow_M^r$ is a binary relation on the configurations of $M$ such that two configurations $\langle Q, m, n\rangle$, $\langle \widehat{Q}, \widehat{m}, \widehat{n}\rangle$ of $M$ are in the in the relation if all of the following conditions hold:

- $Q \neq Q_f$, $r = R(Q)$,
- if $r = +(1, Q')$ for some $Q' \in \mathcal{Q}$ then $\widehat{Q} = Q'$, $\widehat{m} = m + 1$, and $\widehat{n} = n$,
- if $r = +(2, Q')$ for some $Q' \in \mathcal{Q}$ then $\widehat{Q} = Q'$, $\widehat{m} = m$, and $\widehat{n} = n + 1$,
- if $r = -(1, Q_1, Q_2)$ for some $Q_1, Q_2 \in \mathcal{Q}$ then
  if $m = 0$ then $\widehat{Q} = Q_2$, $\widehat{m} = 0$, and $\widehat{n} = n$,
  if $m \geq 1$ then $\widehat{Q} = Q_1$, $\widehat{m} = m - 1$, and $\widehat{n} = n$,
- if $r = -(2, Q_1, Q_2)$ for some $Q_1, Q_2 \in \mathcal{Q}$ then
  if $n = 0$ then $\widehat{Q} = Q_2$, $\widehat{m} = m$, and $\widehat{n} = 0$,
  if $n \geq 1$ then $\widehat{Q} = Q_1$, $\widehat{m} = m$, and $\widehat{n} = n - 1$.

The transition relation of $M$, denoted by $\Rightarrow_M$, is defined as $\bigcup_{r \in R(Q \setminus \{Q_f\})} \Rightarrow^r_M$. We denote the transitive reflexive closure of $\Rightarrow_M$ by $\Rightarrow^*_M$

Let $m, n$ be in $\mathbb{N}$, we say that $M$ terminates on input $(m, n)$ if there exist $\widehat{m}, \widehat{n} \in \mathbb{N}$ such that $\langle Q_0, m, n\rangle \Rightarrow^*_M \langle Q_f, \widehat{m}, \widehat{n}\rangle$ (It follows that there exists an $i \in \mathbb{N}$ and configurations $D_1, \dots, D_i$ of $M$ such that $\langle Q_0, m, n\rangle = D_1 \Rightarrow_M \cdots \Rightarrow_M D_i = \langle Q_f, \widehat{m}, \widehat{n}\rangle$, we call this chain a computation with length $i$).

**Definition 15.** The halting problem for two-counter automaton, denoted by **HALT**, is defined as follows. Given a two-counter automaton $M$.

Does $M$ terminate on input $(0, 0)$?

It is well known that **HALT** is undecidable.

# 3 System P

## 3.1 Definitions

In the following let $\mathcal{V}_P = \{\alpha, a, \beta, b, \dots\}$ be a countably infinite subset of $\mathcal{V}_T$ (of variables). Let $\mathcal{P}_P = \{P, Q, \dots\}$ be a set (of predicate symbols) and $\mathcal{P}$ a ranked set such that $\mathcal{P}^{(0)} = \{\mathbf{false}\}$, $\mathcal{P}^{(2)} = \mathcal{P}_P$, and $\mathcal{P}^{(k)} = \emptyset$ for all $k \in \mathbb{N} \setminus \{0, 2\}$. A first-order logic formula $\varphi$ over $(\mathcal{V}_P, \emptyset, \mathcal{P})$ is an

**atomic formula** if $\varphi = \mathbf{false}$ or $\varphi = P(a, b)$ for some $P \in \mathcal{P}_P$ and $a, b \in \mathcal{V}_P$.

**universal formula** if $\varphi = \forall\vec{\alpha}(A_1 \to A_2 \to \cdots \to A_n)$ for some $n \in \mathbb{N}$ and where $A_i$ is an atomic formula for $i \in \{1, \dots, n\}$, $A_i \neq \mathbf{false}$ for $i \in \{1, \dots, n-1\}$ and for each $\alpha \in \mathrm{FV}(\varphi) \cap \mathrm{FV}(A_n)$ there exists an $i \in \{1, \dots, n-1\}$ such that $\alpha \in \mathrm{FV}(A_i)$.

**existential formula** if there exits an $n \in \mathbb{N}$, atomic formulas $A_i \neq \mathbf{false}$ for $i \in \{1, \dots, n\}$ such that $\varphi = \forall\vec{\alpha}(A_1 \to A_2 \to \cdots \to A_{n-1} \to \forall\beta(A_n \to \mathbf{false}) \to \mathbf{false})$.

The set of formulas of System $\mathbf{P}$ (= set of $\mathbf{P}$-formulas) over $\mathcal{V}_P$ and $\mathcal{P}_P$ is the set of all first-order formulas in $\mathcal{L}_{(\mathcal{V}_P, \emptyset, \mathcal{P})}$ that are either an atomic, universal or existential formula.

**Definition 16.** A finite set of $\mathbf{P}$-formulas $\Gamma$ is called $\mathbf{P}$-basis, or basis if it is clear from the context whether a $\mathbf{P}$-basis or a $\boldsymbol{\lambda}\mathbf{2}$-basis is meant.

For a $\mathbf{P}$-basis $\Gamma$ and a $\mathbf{P}$-formula $A$ we will abbreviate $\Gamma \cup \{A\}$ to $\Gamma, A$ (c.f. $\boldsymbol{\lambda}\mathbf{2}$-basis).

**Definition 17.** Let $A$ be a $\mathbf{P}$-formula, and $\Gamma$ be a basis. The formula $A$ is a semantic consequence of $\Gamma$ if $A$ can be produced using the following deduction rules.

$$
\begin{array}{lll}
\text{(Axiom)} & \Gamma, A \vdash A & \\[2ex]
(\rightarrow\text{-Introduction}) & \dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} & \\[2ex]
(\rightarrow\text{-Elimination}) & \dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} & \\[2ex]
(\forall\text{-Introduction}) & \dfrac{\Gamma \vdash B}{\Gamma \vdash \forall\alpha B} & \alpha \notin \mathrm{FV}(\Gamma) \\[2ex]
(\forall\text{-Elimination}) & \dfrac{\Gamma \vdash \forall\alpha B}{\Gamma \vdash B\,[\alpha := b]} & b \in \mathcal{V}_P
\end{array}
$$

We define a more general consequence relation in which we demand that **false** is interpreted with $\bot$. In this relation existential formulas will behave like the name suggests. Formally:

**Definition 18.** Let $\Gamma$ be a basis. The $\mathbf{P}$-formula $A$ is a <u>sematic consequence with falsity</u> of $\Gamma$, denoted by $\Gamma \vdash_{\mathrm{f}} A$, if for every interpretation $I$

$$I \models \Gamma \text{ and } \mathbf{false}^I = \bot \text{ implies } I \models A.$$

This allows us to add the following deduction rule.

$$
(\exists\text{-Introduction}) \quad \frac{\Gamma, A\,[\alpha := a] \vdash_{\mathrm{f}} B}{\Gamma, A' := \forall\alpha(A \to \mathbf{false}) \to \mathbf{false} \vdash_{\mathrm{f}} B} \quad a \notin FV(\Gamma, A', B)
$$

*Proof.* Let $I = (\Delta, \cdot^I, \omega)$ be a model of $\Gamma, \forall\alpha(A \to \mathbf{false}) \to \mathbf{false}$ with $\mathbf{false}^I = \bot$ and $a \in \mathcal{V}_P$ a variable such that $a \notin FV(\Gamma, A', B)$.

$$
\begin{aligned}
I \models \Gamma, \forall\alpha(A \to \mathbf{false}) \to \mathbf{false} &\Rightarrow I \models \forall\alpha(A \to \mathbf{false}) \to \mathbf{false} \\
&\Rightarrow (\forall\alpha(A \to \mathbf{false}))^I \to \mathbf{false}^I \\
&\Rightarrow (\forall\alpha(A \to \mathbf{false}))^I \to \bot \\
&\Rightarrow \neg(\forall\alpha(A \to \mathbf{false}))^I \\
&\Rightarrow \neg(\forall d \in \Delta\colon (A \to \mathbf{false})^{I[\alpha \mapsto d]}) \\
&\Rightarrow \exists d \in \Delta\colon \neg(A^{I[\alpha \mapsto d]} \to \mathbf{false}^{I[\alpha \mapsto d]}) \\
&\Rightarrow \exists d \in \Delta\colon \neg(A^{I[\alpha \mapsto d]} \to \bot) \\
&\Rightarrow \exists d \in \Delta\colon \neg(\neg A^{I[\alpha \mapsto d]}) \\
&\Rightarrow \exists d \in \Delta\colon A^{I[\alpha \mapsto d]}
\end{aligned}
$$

Together with $a \notin FV(\Gamma, A')$, it follows that $I\,[a \mapsto d]$ is a model of $\Gamma, A\,[\alpha := a]$. Which implies $I\,[a \mapsto d] \models B$. Since $a$ is not free in $B$ we conclude that $I$ is also a model of $B$. □

**Definition 19.** The problem to decide whether a given set of **P**-formulas is consistent, denoted by **CONS**, is defined as follows. Given a set of **P**-formulas $\Gamma$.

$$\text{Does } \Gamma \vdash \textbf{false} \text{ not hold?}$$

## 3.2 CONS is undecidable

We will show that **HALT** $\leq$ **CONS** then the undecidability of **CONS** directly follows from the undecidability of **HALT**. For a given two-counter automaton $M$ we will effectively construct a **P**-basis $\Gamma_M$ such that

$$M \text{ terminates on input } (0,0) \qquad \text{iff} \qquad \Gamma_M \vdash \textbf{false} \text{ holds in System } \mathbf{P}.$$

Let $M = (\mathcal{Q}, Q_0, Q_f, R)$ be a two-counter automaton, w.l.o.g. $S, P, R_1, R_2, E, D \notin \mathcal{Q}$. In the following we will consider **P**-formulas over $\mathcal{V}_P$ and $\mathcal{P}_P$, where $\mathcal{P}_P = \mathcal{Q} \uplus \{S, P, R_1, R_2, E, D\}$. We will abbreviate $P(a,a)$ to $P(a)$, note that this way we can use binary predicate symbols as unary ones.

Intuitively $Q(a)$ stands for "$a$ is in state $Q$", $R_i(a,m)$ stands for "in $a$ the value of register i is $m$" for $i \in \{1,2\}$, $S(a,b)$ states that "$b$ is a successor of $a$", $P(a,b)$ states that "$b$ is a predecessor of $a$", $E(a)$ marks "$a$ as the end of chain", and $D(a)$ states that "$a$ is not the end of a chain".

For a configuration $C = \langle Q, m, n\rangle$ of $M$ we define a set of **P**-formulas $\Gamma_C$. It contains the following formulas:

- $Q(a)$
- $R_1(a, a_0), P(a_{i-1}, a_i)$ for $i \in \{1, \ldots, m\}$
- $R_2(a, b_0), P(b_{i-1}, b_i)$ for $i \in \{1, \ldots, n\}$
- $D(a), D(a_i), D(b_j)$ for $i \in \{0, \ldots, m-1\}$ and $j \in \{0, \ldots, n-1\}$
- $E(a_m), E(b_n)$

Next we need sets of **P**-formulas for all possible transitions. For every $Q \in \mathcal{Q} \setminus \{Q_f\}$ and $r \in \mathcal{R}_Q$ we define $\Gamma_{Q,r}$. If $r = +(1, Q')$ for some $Q' \in \mathcal{Q}$ then $\Gamma_{Q,+(1,Q')}$ contains the following formulas:

- $\forall \alpha\beta(Q(\alpha) \to S(\alpha,\beta) \to Q'(\beta))$
  change of state
- $\forall \alpha\beta\gamma\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to R_1(\beta,\delta) \to P(\delta,\gamma))$
  increment register 1

- $\forall\alpha\beta\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\beta,\delta) \to D(\delta))$
  prevent zero in register 1

- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_2(\alpha,\gamma) \to R_2(\beta,\gamma))$
  do not change the value register 2

If $r = -(1, Q_1, Q_2)$ for some $Q_1, Q_2 \in \mathcal{Q}$ then $\Gamma_{Q,-(1,Q_1,Q_2)}$ contains the following formulas:

- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to E(\gamma) \to Q_2(\beta))$
  jump to $Q_2$ if register 1 is zero

- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to E(\gamma) \to R_1(\beta,\gamma))$
  if register 1 is zero it stays zero

- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to Q_1(\beta))$
  change state to $Q_1$ if register 1 is greater zero

- $\forall\alpha\beta\gamma\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to P(\gamma,\delta) \to R_1(\beta,\delta))$
  decrement register 1

- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_2(\alpha,\gamma) \to R_2(\beta,\gamma))$
  do not change register 2 in both cases

For $r = +(2, Q')$ for some $Q' \in \mathcal{Q}$ or $r = -(2, Q_1, Q_2)$ for some $Q_1, Q_2 \in \mathcal{Q}$ the sets $\Gamma_{Q,r}$ are defined analogously.

We also need a set $\Gamma_1$ to ensure that our representation works correctly. The following formula are in $\Gamma_1$:

- $\forall\alpha\beta(S(\alpha,\beta) \to D(\beta))$
  no successor is the end of a chain

- $\forall\alpha(D(\alpha) \to \forall\beta(R_1(\alpha,\beta) \to \mathbf{false}) \to \mathbf{false})$
  every element that represents a configuration has a value for register 1

- $\forall\alpha(D(\alpha) \to \forall\beta(R_2(\alpha,\beta) \to \mathbf{false}) \to \mathbf{false})$
  every element that represents a configuration has a value for register 2

- $\forall\alpha(\forall\beta(S(\alpha,\beta) \to \mathbf{false}) \to \mathbf{false})$
  every element has a successor

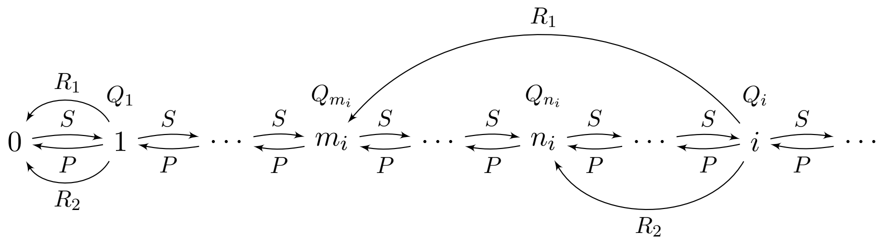
We define $\Gamma_{\overline{M}}$ as $\bigcup_{Q\in\mathcal{Q}\setminus\{Q_f\}} \Gamma_{Q,R(Q)} \cup \{\forall\alpha(Q_f(\alpha) \to \mathbf{false})\} \cup \Gamma_1$. Finally we can define $\Gamma_M$ as $\Gamma_{C_1} \cup \Gamma_{\overline{M}}$, where $C_1 = \langle Q_0, 0, 0\rangle$ is the initial configuration.

**Claim 20.**

$$\Gamma_M \vdash \textit{\textbf{false}} \textit{ holds in system } P \qquad \Longrightarrow \qquad M \textit{ terminates on input } (0,0)$$

*Proof.* Assume $M$ does not terminate then there is an infinite chain $C_1 \Rightarrow_M C_2 \Rightarrow_M C_3 \Rightarrow_M \ldots$ ($C_i = \langle Q_i, m_i, n_i\rangle$ for $i \in \mathbb{N}^+$). Now we construct a model of $\Gamma_M$ which interprets **false** with $\perp$ this contradicts $\Gamma_M \vdash$ **false**.

To illustrate the idea we will use a graphical notation for an interpretation $I$. By $d_1 \xrightarrow{\mathrm{R}} d_2$ we say that $(d_1, d_2) \in R^I$. And we use $\overset{\mathrm{P}}{d}$ to say that $(d, d) \in P^I$ for predicate symbols that are used as unary predicate symbols. As domain for our interpretation we will use the natural numbers. Every number will have two tasks: firstly it will represent itself as a possible value for register 1 or 2 and secondly every number $i$ greater than zero will also represent the $i^{\text{th}}$ configuration of our infinite computation. Now the idea for our model of $\Gamma_M$ looks like this:




We have $0 \in E^I$ and all other numbers are in $D^I$.

Here is the more formal definition of our model $I = (\mathbb{N}, \cdot^I, \omega)$.

$$
\begin{array}{lll}
P^I = \{(i+1, i) \mid i \in \mathbb{N}\} & R_1^I = \{(i, m_i) \mid i \in \mathbb{N}\} & R_2^I = \{(i, n_i) \mid i \in \mathbb{N}\} \\
Q^I = \{(i, i) \mid i \in \mathbb{N}^+, Q = Q_i\} & D^I = \{(i, i) \mid i \in \mathbb{N}^+\} & E^I = \{(0, 0)\} \\
S^I = \{(i, i+1) \mid i \in \mathbb{N}\} & \textbf{false}^I = \perp &
\end{array}
$$

$$
a^I = 1 \qquad\qquad a_0^I = 0 \qquad\qquad b_0^I = 0
$$

Since there are no free variables in $\Gamma_M$ we can just set $\omega(x) = 0$ for every $x \in \mathcal{V}_P$. It is easy to see that $I$ is indeed a model of $\Gamma_M$. $\square$

We proof the other direction by induction on the length of the computation. But to be able to use the induction hypothesis we need a slightly more general statement.

**Claim 21.** *Let $C = \langle Q, m, n\rangle$ be a configuration of $M$. If a final configuration (i.e. a configuration $\langle Q_f, \widehat{m}, \widehat{n}\rangle$ for some $\widehat{n}, \widehat{m} \in \mathbb{N}$) is reachable from $C$ then $\Gamma_C \cup \Gamma_{\overline{M}} \vdash$ **false**.*

*Proof.* By induction on the length $i$ of the computation.

Induction Base: $i = 0$

Since a final configuration is reachable in 0 steps $C$ must be this final configuration. So $C = \langle Q_f, m, n\rangle$ for some $n, m \in \mathbb{N}$. Hence, $Q_f(a)$ is in $\Gamma_C$ for some $a \in \mathcal{V}_P$ and $\forall\alpha(Q_f(\alpha) \to \textbf{false})$ is in $\Gamma_{\overline{M}}$, we can easily deduce false.

$$\dfrac{\dfrac{\Gamma_C \cup \Gamma_{\overline{M}} \vdash \forall\alpha(Q_f(\alpha) \to \textbf{false})}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash Q_f(a) \to \textbf{false}} \qquad \Gamma_C \cup \Gamma_{\overline{M}} \vdash Q_f(a)}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash \textbf{false}}$$

Induction Step: $i = i' + 1$

Since $I \models$ **false** holds trivially if $I$ interprets **false** with $\top$ we only need to consider models of $\Gamma_C \cup \Gamma_{\overline{M}}$ that interpret **false** with $\perp$ (note that there are no such models if $M$ terminates which is exactly what we want to proof). As result of this observation we can use the $\exists$-Introduction rule.

From the fact that a final configuration is reachable from $C$ in $i$ steps we can deduce that there exists a configuration $D = \langle \widehat{Q}, \widehat{m}, \widehat{n} \rangle$ such that $C \Rightarrow_M^r D$ for some $r \in \mathcal{R}_\mathcal{Q}$ and a final configuration is reachable from $D$ in $i'$ steps. We also know that $C = \langle Q, m, n \rangle$ for some $Q \in \mathcal{Q} \setminus \{Q_f\}$ and some $m, n \in \mathbb{N}$. The set $\Gamma_C$ contains the formulas:

$R_1(a, a_0)$, $P(a_{i-1}, a_i)$ and $D(a_{i-1})$ for $i \in \{1, \dots, n\}$,

$R_2(a, b_0)$, $P(b_{i-1}, b_i)$ and $D(b_{i-1})$ for $i \in \{1, \dots, m\}$,

$Q(a)$, $D(a)$, $E(a_n)$ and $E(b_m)$.

And $\Gamma_D$ contains the formulas:

$R_1(\widehat{a}, \widehat{a}_0)$, $P(\widehat{a}_{i-1}, \widehat{a}_i)$ and $D(\widehat{a}_{i-1})$ for $i \in \{1, \dots, \widehat{n}\}$,

$R_2(\widehat{a}, \widehat{b}_0)$, $P(\widehat{b}_{i-1}, \widehat{b}_i)$ and $D(\widehat{b}_{i-1})$ for $i \in \{1, \dots, \widehat{m}\}$,

$Q(\widehat{a})$, $D(\widehat{a})$, $E(\widehat{a}_{\widehat{n}})$ and $E(\widehat{b}_{\widehat{m}})$.

The basic idea is to deduce $\Gamma_D$ from $\Gamma_C \cup \Gamma_{\overline{M}}$ and then apply the induction hypothesis to $\Gamma_D \cup \Gamma_{\overline{M}}$.

$$\dfrac{\dfrac{\text{Induction Hypothesis}}{\Gamma_C \cup \Gamma_{\overline{M}} \cup \Gamma_D \vdash_\mathrm{f} \textbf{false}} \qquad \Gamma_C \cup \Gamma_{\overline{M}} \vdash_\mathrm{f} \Gamma_D}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash_\mathrm{f} \textbf{false}}$$

We achieve this by looking at the four possible cases for the type of the rule $r$. We will only consider the cases $r = +(1, Q')$ and $r = -(1, Q_1, Q_2)$, the two remaining cases $r = +(2, Q')$ and $r = -(2, Q_1, Q_2)$ follow by exchanging the roles of register 1 and register 2 in the first two cases.

First we need a new free variable representing the configuration $D$. Also the value in register 2 does not change, because in both cases we are only concerned with register 1. For the succeeding tableau proofs we will abbreviate **false** by **f** and we will drop $\Gamma_C \cup \Gamma_{\overline{M}}$ and only write new formulas on the left side of $\vdash_\mathrm{f}$.

We first introduce a new variable representing $D$ (let $b \in \mathcal{V}_P \setminus \mathrm{FV}(\Gamma_C \cup \Gamma_{\overline{M}})$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{S(a,b), D(b) \vdash_{\mathrm{f}} \mathbf{f}}}{S(a,b) \vdash_{\mathrm{f}} D(b) \to \mathbf{f}} \quad \dfrac{\dfrac{S(a,b) \vdash_{\mathrm{f}} \forall\alpha\beta(S(\alpha,\beta) \to D(\beta))}{S(a,b) \vdash_{\mathrm{f}} S(a,b) \to D(b)} \quad S(a,b) \vdash_{\mathrm{f}} S(a,b)}{S(a,b) \vdash_{\mathrm{f}} D(b)}}{S(a,b) \vdash_{\mathrm{f}} \mathbf{f}}}{\forall\beta(S(a,\beta) \to \mathbf{f}) \to \mathbf{f} \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} (\forall\beta(S(a,\beta) \to \mathbf{f}) \to \mathbf{f}) \to \mathbf{f}} \quad \dfrac{\vdash_{\mathrm{f}} \forall\alpha(\forall\beta(S(\alpha,\beta) \to \mathbf{f}) \to \mathbf{f})}{\vdash_{\mathrm{f}} \forall\beta(S(a,\beta) \to \mathbf{f}) \to \mathbf{f}}}{\vdash_{\mathrm{f}} \mathbf{f}}$$

Since register 2 should not change we need $R_2(b, b_0)$. Again we will just drop $S(a,b)$ and $D(b)$ on the left side for comprehensibility.

$$\dfrac{\dfrac{\dfrac{\vdots}{R_2(b,b_0) \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} R_2(b,b_0) \to \mathbf{f}} \quad \dfrac{\dfrac{\dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_2(\alpha,\gamma) \to R_2(\beta,\gamma))}{\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to R_2(a,b_0) \to R_2(b,b_0)} \quad \vdash_{\mathrm{f}} Q(a)}{\vdash_{\mathrm{f}} S(a,b) \to R_2(a,b_0) \to R_2(b,b_0)} \quad \vdash_{\mathrm{f}} S(a,b)}{\vdash_{\mathrm{f}} R_2(a,b_0) \to R_2(b,b_0)} \quad \vdash_{\mathrm{f}} R_2(a,b_0)}{\vdash_{\mathrm{f}} R_2(b,b_0)}}{\vdash_{\mathrm{f}} \mathbf{f}}$$

For the case that $\boldsymbol{r = +(1, Q')}$, we have that $\widehat{Q} = Q'$, $\widehat{m} = m + 1$, and $\widehat{n} = n$. So we need to increment register 1 and ensure that the state of $b$ is $Q'$.

$$\dfrac{\dfrac{\dfrac{\vdots}{Q'(b) \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} Q'(b) \to \mathbf{f}} \quad \dfrac{\dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta(Q(\alpha) \to S(\alpha,\beta) \to Q'(\beta))}{\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to Q'(b)} \quad \vdash_{\mathrm{f}} Q(a)}{\vdash_{\mathrm{f}} S(a,b) \to Q'(b)} \quad \vdash_{\mathrm{f}} S(a,b)}{\vdash_{\mathrm{f}} Q'(b)}}{\vdash_{\mathrm{f}} \mathbf{f}}$$

To increment register 1 we need a new free variable as anchor for register 1 (let $d \in \mathcal{V}_P \setminus \mathrm{FV}(\Gamma_C \cup \Gamma_{\overline{M}})$ and $d \neq b$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R_1(b,d) \vdash_{\mathrm{f}} \mathbf{f}}}{\forall\beta(R_1(b,\beta) \to \mathbf{f}) \to \mathbf{f} \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} (\forall\beta(R_1(b,\beta) \to \mathbf{f}) \to \mathbf{f}) \to \mathbf{f}} \quad \dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha(D(\alpha) \to \forall\beta(R_1(\alpha,\beta) \to \mathbf{f}) \to \mathbf{f})}{\vdash_{\mathrm{f}} D(b) \to \forall\beta(R_1(b,\beta) \to \mathbf{f}) \to \mathbf{f}} \quad \vdash_{\mathrm{f}} D(b)}{\vdash_{\mathrm{f}} \forall\beta(R_1(b,\beta) \to \mathbf{f}) \to \mathbf{f}}}{\vdash_{\mathrm{f}} \mathbf{f}}$$

Now we need to connect $d$ with $a_0$ (the anchor of $a$ for register 1).

$$
\dfrac{\dfrac{\dfrac{\vdots}{P(d,a_0)\vdash_{\mathrm{f}}\mathbf{f}}}{\vdash_{\mathrm{f}} P(d,a_0)\to\mathbf{f}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta\gamma\delta(Q(\alpha)\to S(\alpha,\beta)\to R_1(\alpha,\gamma)\to R_1(\beta,\delta)\to P(\delta,\gamma))}{\vdash_{\mathrm{f}} Q(a)\to S(a,b)\to R_1(a,a_0)\to R_1(b,d)\to Q'(b)} \quad \vdash_{\mathrm{f}} Q(a)}{\vdash_{\mathrm{f}} S(a,b)\to R_1(a,a_0)\to R_1(b,d)\to Q'(b)} \quad \vdash_{\mathrm{f}} S(a,b)}{\vdash_{\mathrm{f}} R_1(a,a_0)\to R_1(b,d)\to Q'(b)} \quad \vdash_{\mathrm{f}} R_1(a,a_0)}{\vdash_{\mathrm{f}} R_1(b,d)\to Q'(b)} \quad \vdash_{\mathrm{f}} R_1(b,d)}{\vdash_{\mathrm{f}} P(d,a_0)}}{\vdash_{\mathrm{f}}\mathbf{f}}
$$

At last we have to make sure that we do not get an artificial zero. We achieve this by deducing $D(d)$.

$$
\dfrac{\dfrac{\dfrac{\vdots}{D(d)\vdash_{\mathrm{f}}\mathbf{f}}}{\vdash_{\mathrm{f}} D(d)\to\mathbf{f}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta\delta(Q(\alpha)\to S(\alpha,\beta)\to R_1(\beta,\delta)\to D(\delta))}{\vdash_{\mathrm{f}} Q(a)\to S(a,b)\to R_1(b,d)\to D(d)} \quad \vdash_{\mathrm{f}} Q(a)}{\vdash_{\mathrm{f}} S(a,b)\to R_1(b,d)\to D(d)} \quad \vdash_{\mathrm{f}} S(a,b)}{\vdash_{\mathrm{f}} R_1(b,d)\to D(d)} \quad \vdash_{\mathrm{f}} R_1(b,d)}{\vdash_{\mathrm{f}} D(d)}}{\vdash_{\mathrm{f}}\mathbf{f}}
$$

Now we already have deduced $\Gamma_D$, to see why define $\widehat{a} := b$, $\widehat{b_i} := b_i$ for $i \in \{0,\ldots,m\}$, $\widehat{a}_0 := d$, and $\widehat{a}_{i+1} := a_i$ for $i \in \{0,\ldots,n\}$. Hence we can deduce **false** by induction hypothesis.

The other case, that $\boldsymbol{r = -(Q, 1, Q_1, Q_2)}$, has to be split into two cases again. If $\boldsymbol{m = 0}$ then $\widehat{Q} = Q_2$, $\widehat{m} = 0$, and $\widehat{n} = n$. We only need to ensure that the successor state is $Q_2$ and that register 1 is still zero.

$$
\dfrac{\dfrac{\dfrac{\vdots}{Q_2(b)\vdash_{\mathrm{f}}\mathbf{f}}}{\vdash_{\mathrm{f}} Q_2(b)\to\mathbf{f}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta\gamma(Q(\alpha)\to S(\alpha,\beta)\to R_1(\alpha,\gamma)\to E(\gamma)\to Q_2(\beta))}{\vdash_{\mathrm{f}} Q(a)\to S(a,b)\to R_1(a,a_0)\to E(a_0)\to Q_2(b)} \quad \vdash_{\mathrm{f}} Q(a)}{\vdash_{\mathrm{f}} S(a,b)\to R_1(a,a_0)\to E(a_0)\to Q_2(b)} \quad \vdash_{\mathrm{f}} S(a,b)}{\vdash_{\mathrm{f}} R_1(a,a_0)\to E(a_0)\to Q_2(b)} \quad \vdash_{\mathrm{f}} R_1(a,a_0)}{\vdash_{\mathrm{f}} E(a_0)\to Q_2(b)} \quad \vdash_{\mathrm{f}} E(a_0)}{\vdash_{\mathrm{f}} Q_2(b)}}{\vdash_{\mathrm{f}}\mathbf{f}}
$$

Register 1 stays zero.

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\UnaryInfC{$R_1(b,a_0) \vdash_{\mathrm{f}} \mathbf{f}$}
\UnaryInfC{$\vdash_{\mathrm{f}} R_1(b,a_0) \to \mathbf{f}$}
\AxiomC{$\vdash_{\mathrm{f}} \forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to E(\gamma) \to R_1(\beta,\gamma))$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to R_1(a,a_0) \to E(a_0) \to R_1(b,a_0)$}
\AxiomC{$\vdash_{\mathrm{f}} Q(a)$}
\BinaryInfC{$\vdash_{\mathrm{f}} S(a,b) \to R_1(a,a_0) \to E(a_0) \to R_1(b,a_0)$}
\AxiomC{$\vdash_{\mathrm{f}} S(a,b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(a,a_0) \to E(a_0) \to R_1(b,a_0)$}
\AxiomC{$\vdash_{\mathrm{f}} R_1(a,a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} E(a_0) \to R_1(b,a_0)$}
\AxiomC{$\vdash_{\mathrm{f}} E(a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(b,a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} \mathbf{f}$}
\end{prooftree}
$$

If we define $\widehat{a} := b$, $\widehat{b}_i := b_i$ for $i \in \{0,\ldots,m\}$, and $\widehat{a}_0 := a_0$ then it is clear that we have deduced all formulas required for $\Gamma_D$. So we can use the induction hypothesis to deduce **false**.

In the last case $\boldsymbol{m > 0}$, so $\widehat{Q} = Q_1$, $\widehat{m} = m-1$, and $\widehat{n} = n$. First we ensure that $b$ is in state $Q_1$.

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\UnaryInfC{$Q_1(b) \vdash_{\mathrm{f}} \mathbf{f}$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q_1(b) \to \mathbf{f}$}
\AxiomC{$\vdash_{\mathrm{f}} \forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to Q_1(\beta))$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to R_1(a,a_0) \to D(a_0) \to Q_1(b)$}
\AxiomC{$\vdash_{\mathrm{f}} Q(a)$}
\BinaryInfC{$\vdash_{\mathrm{f}} S(a,b) \to R_1(a,a_0) \to D(a_0) \to Q_1(b)$}
\AxiomC{$\vdash_{\mathrm{f}} S(a,b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(a,a_0) \to D(a_0) \to Q_1(b)$}
\AxiomC{$\vdash_{\mathrm{f}} R_1(a,a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} D(a_0) \to Q_1(b)$}
\AxiomC{$\vdash_{\mathrm{f}} D(a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} Q_1(b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} \mathbf{f}$}
\end{prooftree}
$$

Now we decrement register 1 by taking $a_1$ (the predecessor of $a_0$) as anchor of $b$ for register 1.

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\UnaryInfC{$R_1(b,a_1) \vdash_{\mathrm{f}} \mathbf{f}$}
\UnaryInfC{$\vdash_{\mathrm{f}} R_1(b,a_1) \to \mathbf{f}$}
\AxiomC{$\vdash_{\mathrm{f}} \forall\alpha\beta\gamma\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to P(\gamma,\delta) \to R_1(\beta,\delta))$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)$}
\AxiomC{$\vdash_{\mathrm{f}} Q(a)$}
\BinaryInfC{$\vdash_{\mathrm{f}} S(a,b) \to R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)$}
\AxiomC{$\vdash_{\mathrm{f}} S(a,b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)$}
\AxiomC{$\vdash_{\mathrm{f}} R_1(a,a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)$}
\AxiomC{$\vdash_{\mathrm{f}} D(a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} P(a_0,a_1) \to R_1(b,a_1)$}
\AxiomC{$\vdash_{\mathrm{f}} P(a_0,a_1)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(b,a_1)$}
\BinaryInfC{$\vdash_{\mathrm{f}} \mathbf{f}$}
\end{prooftree}
$$

Again it is obvious that we have deduced $\Gamma_D$ ($\widehat{a} := b$, $\widehat{b}_i := b_i$ for $i \in \{0,\ldots,m\}$, and $\widehat{a}_{i-1} := a_i$ for $i \in \{1,\ldots,n\}$). Hence, by induction hypothesis, we can deduce **false**. □

**Lemma 22.**

*$M$ terminates on input $(0,0)$ iff $\Gamma_M \vdash$ **false** holds in system $P$.*

*Proof.* The $\Leftarrow$ directions is proven in Claim 20. And the $\Rightarrow$ direction is a direct consequence of Claim 21 with $C = \langle Q_0, 0, 0\rangle$. □

**Theorem 23.** ***CONS*** *is undecidable.*

*Proof.* Since by Lemma 22 for a given two-counter automaton $M$ we can effectively construct a set of $\mathbf{P}$-formulas $\Gamma_M$ such that $M$ terminates on input $(0,0)$ iff $\Gamma_M$ is not consistent. It follows that **HALT** $\leq$ **CONS**. Since **HALT** is undecidable we have shown that **CONS** is undecidable too. $\square$

## References


[1] H.P. Barendregt, 1993. Lambda Calculi with Types, Handbook of Logic in Computer Science, Volume II, 34-68.