Contents

1 Introduction

2 Basic Definitions

We will denote the set $\{1, \ldots, n\}$ by $[n]$.

2.1 λ-calculus $\lambda 2$

$FV(\Gamma) = \bigcup \{FV(t) \mid (x : t) \in \Gamma\}$
In the following let $\mathcal{V}_T = \{\alpha, \beta, \ldots\}$ be a countable set (of type-variables) and $\mathcal{V}_V = \{x_1, x_2, \ldots\}$ 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 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 $\forall \alpha . t \in \mathrm{T}$.

Definition 2. 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 Λ_T satisfying the following conditions:

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

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:

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

Or is this definition better?

Definition 4. 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 5. A basis is a finite subset of $\mathcal{V}_V \times \Lambda_{\mathrm{T}_{\lambda 2}}$

$\lambda 2$ deduction Rules

(Axiom)	$\Gamma, x : t \vdash x : t$	
(λ-Introduction)	$\dfrac{\Gamma, x : t_1 \vdash e : t_2}{\Gamma \vdash \lambda x.e : t_1 \to t_2}$	
(λ-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)	$\dfrac{\Gamma \vdash e : \forall\alpha.t}{\Gamma \vdash e\,t' : t\,[\alpha := t']}$	$t' \in \mathrm{T}_{\lambda 2}$

2.2 first-order logic

Definition 6. A ranked set is a tuple (Σ, rk), where Σ is a countable set and $rk : \Sigma \to \mathbb{N}$ is a function that maps every symbol from Σ to a natural number (its rank).

If the function rk is understood we will just write Σ instead of (Σ, rk). The set of all elements with a certain rank k in Σ, denoted by $\Sigma^{(k)}$, is defined by $\Sigma^{(k)} := rk^{-1}(k)$. In the following we will write $\Sigma = \{P^{(0)}, Q^{(3)}\}$ to say that $\Sigma = \{P, Q\}$, $rk(P) = 0$, and $rk(Q) = 3$.

In the following let $\mathcal{V} = \{x_0, x_1, \dots\}$ be a countable set (of variables), $\mathcal{F}$ a ranked set (of function symbols), and $\mathcal{P}$ a ranked set (of predicate symbols).

Definition 7. The set of terms over $(\mathcal{V}, \mathcal{F})$, 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 $(\mathcal{V}, \mathcal{F}, \mathcal{P})$, 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 $x \in \mathcal{V}$ and $\varphi \in \mathcal{L}$ then $\exists x\varphi$, $\forall x\varphi \in \mathcal{L}$.

We introduce an additional binary operation $\rightarrow$ on formulas, where for some φ, $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$ the formula $(\varphi \rightarrow \psi)$ is defined as $(\neg\varphi \vee \psi)$.

Definition 8. The variables of a term $t \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$, denoted by $\mathrm{V}(t)$, are defined by:

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

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:

$$\mathrm{FV}(\varphi) = \begin{cases} \mathrm{V}(t_1) \cup \mathrm{V}(t_2) \cup \cdots \cup \mathrm{V}(t_k) & \text{if } \varphi = P(t_1, t_2, \ldots, t_k) \\ \mathrm{FV}(\varphi_1) \cup \mathrm{FV}(\varphi_2) & \text{if } \varphi = \varphi_1 \circ \varphi_2,\ \circ \in \{\wedge, \vee\} \\ \mathrm{FV}(\psi) \setminus \{x\} & \text{if } \varphi = Qx\psi,\ Q \in \{\forall, \exists\} \end{cases}$$

Definition 9. Let x be in $\mathcal{V}$ and $t, t' \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$. The substitution of x by t' in t, denoted by $t\,[x := t']$, is defined as follows:

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

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

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

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

Definition 10. An interpretation I over $(\mathcal{V}, \mathcal{F}, \mathcal{P})$ is a triple $(\Delta, \cdot^I, \omega)$ where

Δ	is a nonempty set (which we call domain),
$\cdot^I$	is a function such that $f^I : \Delta^k \rightarrow \Delta$ 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 $k \in \mathbb{N}$, $f \in \mathcal{P}^{(k)}$
ω	is a function from $\mathcal{V}$ to Δ.

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

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

Definition 11. 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:

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

Definition 12. Let $I = (\Delta, \cdot^I, \omega)$ be an interpretation and φ a formula the interpretation of φ under I, denoted by φ^I, is defined recursively as follows:

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

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

Definition 13. Let Γ be a finite set of first-oder formulas.

We say that an interpretation I is a model of Γ if $I \models \psi$ for every ψ in Γ.

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

The free variables of Γ, 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 try to decrement a register and jump of the register is already zero. Formally:

Definition 14. A deterministic two-counter automaton is a 4-tuple $M = (Q, q_0, q_f, R)$,

where Q is a finite set (of states),
q_0 is in Q (the initial state),
q_f is in Q (the final state), and
R is a function from $Q \setminus \{q_f\}$ to $\mathcal{R}_Q$,
where $\mathcal{R}_Q = \{+(i, q') \mid i \in \{0,1\}, q' \in Q\}$
$\cup\{-(i, q_1, q_2) \mid i \in \{0,1\}, q_1, q_2 \in Q\}$

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

- $q \neq q_f$, $r = R(q)$,
- if $r = +(0, p)$ for some $p \in Q$ then $q' = p$, $m' = m + 1$, and $n' = n$,
- if $r = +(1, p)$ for some $p \in Q$ then $q' = p$, $m' = m$, and $n' = n + 1$,
- if $r = -(0, p_1, p_2)$ for some $p_1, p_2 \in Q$ then
  if $m = 0$ then $q' = p_2$, $m' = 0$, and $n' = n$,
  if $m \geq 1$ then $q' = p_1$, $m' = m - 1$, and $n' = n$,
- if $r = -(1, p_1, p_2)$ for some $p_1, p_2 \in Q$ then
  if $n = 0$ then $q' = p_2$, $m' = m$, and $n' = 0$,
  if $n \geq 1$ then $q' = p_1$, $m' = m$, and $n' = n - 1$.

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 $m', n' \in \mathbb{N}$ such that $\langle q_0, m, n\rangle \Rightarrow^*_M \langle q_f, m', n'\rangle$.

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 set (of variables) and $\mathcal{P}_P = \{\mathit{false}^{(0)}, P^{(2)}, Q^{(2)}, \dots\}$ a ranked set (of predicate symbols) such that $\mathcal{P}^{(0)}_P = \{\mathit{false}\}$, $\mathcal{P}^{(2)}_P = \{P, Q, \dots\}$ is a countably infinite set, and $\mathcal{P}^{(k)}_P = \emptyset$ for all $k \in \mathbb{N} \setminus \{0, 2\}$. A first-order logic formula φ over $(\mathcal{V}_P, \emptyset, \mathcal{P}_P)$ is an

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

universal formula if $\varphi = \forall\overrightarrow{\alpha}(A_1 \to A_2 \to \dots \to A_n)$ where A_i is an atomic formula for $i \in [n]$, $A_i \neq \mathit{false}$ for $i \in [n-1]$ and for each $\alpha \in \mathrm{FV}(\varphi) \cap \mathrm{FV}(A_n)$ there exists an $i \in [n-1]$ such that $\alpha \in \mathrm{FV}(A_i)$.

existential formula if there exits $n \geq 0$, atomic formulas $A_i \neq \mathit{false}$ for $i \in [n]$ such that $\varphi = \forall \vec{\alpha}(A_1 \to A_2 \to \cdots \to A_{n-1} \to \forall \beta(A_n \to \mathit{false}) \to \mathit{false})$.

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

Deduction Rules

(Axiom)	$\Gamma, A \vdash A$	
$(\to$ -Introduction)	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$	
$(\to$ -Elimination)	$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$	
($\forall$-Introduction)	$\dfrac{\Gamma \vdash B}{\Gamma \vdash \forall \alpha B}$	$\alpha \notin \mathrm{FV}(\Gamma)$
($\forall$-Elimination)	$\dfrac{\Gamma \vdash \forall \alpha B}{\Gamma \vdash B\,[\alpha := b]}$	$b \in \mathcal{V}_P$

An Interpretation I of a P formula is a tuple $I = (\Delta, \cdot^I)$ where Δ is a set (called domain), $P^I \subseteq \Delta^k$ and $\alpha^I \in \Delta \ldots$

If we interpret *false* with the logical constant false $(\bot)$ (denoted by $\vdash_f$) we can add a new deduction rule.

($\exists$-Introduction)	$\dfrac{\Gamma, A\,[\alpha := a] \vdash_f B}{\Gamma, \forall \alpha(A \to \mathit{false}) \to \mathit{false} \vdash_f B}$	$a \notin FV(\Gamma, A, B)$

Proof. Let $I = (\Delta, \cdot^I)$ be a model of $\Gamma, \forall \alpha(A \to \mathit{false}) \to \mathit{false}$ with $\mathit{false}^I = \bot$.

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

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 16. 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 Γ.

$$\text{Does } \Gamma \vdash \textit{false} \text{ not hold.}$$

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 set of **P**-formulas Γ_M such that Γ_C :

- $Q(a)$
- $R_1(a, a_0), P(a_{i-1}, a_i)$ 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 $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$
- $E(a_m), E(b_n)$

$+(Q, 1, Q')$:

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

$-(Q, 1, Q_1, Q_2)$:

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

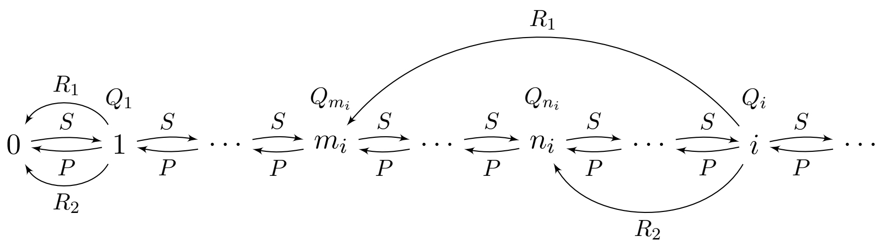
- $\forall\alpha\beta\gamma\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to P(\gamma,\delta) \to R_1(\beta,\delta))$
  decrement register 1
- $\forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_2(\alpha,\gamma) \to R_2(\beta,\gamma))$
  do not change register 2

Claim 17.

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

Proof. Assume M does not terminate then there is an infinite chain $C_1 \Rightarrow_M C_2 \Rightarrow_M C_3 \Rightarrow_M \ldots$ ($C_i = \langle Q_i, m_i, n_i\rangle$) Now we construct a model of Γ_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 $d_1 \overset{\text{R}}{\to} d_2$ we say that $(d_1, d_2) \in R^I$. And we use $\overset{\text{P}}{d}$ to say that $d \in P^I$ for unary predicate symbols. Now the idea for our model of Γ_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 $I = (\mathbb{N}, \cdot^I)$.

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

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

□

Claim 18. *Let C be a configuration of M. If a final state is reachable from C then* $\Gamma_C \cup \Gamma \vdash$ false*.*

Proof. By induction on the length of the computation. For the tableau proofs we will abbreviate *false* by f.
Induction Base trivial ...
Induction Step
$C \Rightarrow^r_M D$

We need to make a case distinction on the rule r.

Case $r = +(Q, 1, Q')$

Basic idea:

$$\dfrac{\dfrac{IH}{\Gamma_C \cup \Gamma \cup \Gamma_D \vdash f} \qquad \dfrac{}{\Gamma_C \cup \Gamma \vdash \Gamma_D}}{\Gamma_C \cup \Gamma \vdash f}$$

Since $I \models \mathit{false}$ holds trivially if I interprets *false* with $\top$ we only need to consider models (note that there are none if M terminates which is exactly what we want to proof) of $\Gamma_C \cup \Gamma$ that interpret *false* with $\perp$ (so we can use our new deduction rule).
We will just drop $\Gamma_C \cup \Gamma$ and only write new formulas on the left side.
We first introduce the new variables needed for Γ_D (let $b, d \in \mathcal{V}_P \backslash \mathrm{FV}(\Gamma_C \cup \Gamma)$). Intuitively b will represent the successor state and d will be the anchor for register one.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{S(a,b), D(b) \vdash_f f}}{S(a,b) \vdash_f D(b) \to f} \qquad \dfrac{\dfrac{S(a,b) \vdash_f \forall\alpha\beta(S(\alpha,\beta) \to D(\beta))}{S(a,b) \vdash_f S(a,b) \to D(b)} \quad S(a,b) \vdash_f S(a,b)}{S(a,b) \vdash_f D(b)}}{S(a,b) \vdash_f f}}{\dfrac{\forall\beta(S(a,\beta) \to f) \to f \vdash_f f}{\vdash_f (\forall\beta(S(a,\beta) \to f) \to f) \to f}} \qquad \dfrac{\vdash_f \forall\alpha(\forall\beta(S(\alpha,\beta) \to f) \to f)}{\vdash_f \forall\beta(S(a,\beta) \to f) \to f}}{\vdash_f f}$$

The formula $R_1(b,d)$ can be acquired in a similar way. Again we will just drop $S(a,b)$ and $D(b)$ on the left side for comprehensibility.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R_1(b,d) \vdash_f f}}{\forall\beta(R_1(b,\beta) \to f) \to f \vdash_f f}}{\vdash_f (\forall\beta(R_1(b,\beta) \to f) \to f) \to f} \qquad \dfrac{\dfrac{\vdash_f \forall\alpha(D(\alpha) \to \forall\beta(R_1(\alpha,\beta) \to f) \to f)}{\vdash_f D(b) \to \forall\beta(R_1(b,\beta) \to f) \to f} \quad \vdash_f D(b)}{\vdash_f \forall\beta(R_1(b,\beta) \to f) \to f}}{\vdash_f f}$$

Now we have all the new free variables we need and we continue by ensuring that these variables fulfill all the formulas in Γ_D.

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

Starting from $Q'(b) \vdash_f \mathit{false}$ we can connect d and a_0.

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

For register one we still need $D(d)$.

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

Since register two should not change we only need $R_2(b,b_0)$.

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

Now we have Γ_C (Since $P(a_{i-1},a_i)$ is already in Γ_D) and can deduce *false* by induction hypothesis.
<u>Case $r = -(Q,1,Q_1,Q_2)$</u>
<u>$r_1 = 0$</u>

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

r_1 stays zero

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

$\underline{r_1 \geq 1}$
new state Q_1

$$\dfrac{\dfrac{\dfrac{\vdots}{Q_1(b) \vdash_f f}}{\vdash_f Q_1(b) \to f} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash_f \forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to Q_1(\beta))}{\vdash_f Q(a) \to S(a,b) \to R_1(a,a_0) \to D(a_0) \to Q_1(b)} \quad \vdash_f Q(a)}{\vdash_f S(a,b) \to R_1(a,a_0) \to D(a_0) \to Q_1(b)} \quad \vdash_f S(a,b)}{\vdash_f R_1(a,a_0) \to D(a_0) \to Q_1(b)} \quad \vdash_f R_1(a,a_0)}{\vdash_f D(a_0) \to Q_1(b)} \quad \vdash_f D(a_0)}{\vdash_f Q_1(b)}}{\vdash_f f}$$

decrement r_1

$$\dfrac{\dfrac{\dfrac{\vdots}{R_1(b,a_1) \vdash_f f}}{\vdash_f R_1(b,a_1) \to f} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash_f \forall\alpha\beta\gamma\delta(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to D(\gamma) \to P(\gamma,\delta) \to R_1(\beta,\delta))}{\vdash_f Q(a) \to S(a,b) \to R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)} \quad \vdash_f Q(a)}{\vdash_f S(a,b) \to R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)} \quad \vdash_f S(a,b)}{\vdash_f R_1(a,a_0) \to D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)} \quad \vdash_f R_1(a,a_0)}{\vdash_f D(a_0) \to P(a_0,a_1) \to R_1(b,a_1)} \quad \vdash_f D(a_0)}{\vdash_f P(a_0,a_1) \to R_1(b,a_1)} \quad \vdash_f P(a_0,a_1)}{\vdash_f R_1(b,a_1)}}{\vdash_f f}$$

$\square$

Lemma 19.

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

Proof. The $\Leftarrow$ directions follows directly from Claim 17. And the $\Rightarrow$ direction is a direct consequence of Claim 18 with $C = \langle q_0, 0, 0 \rangle$. $\square$

Theorem 20. ***CONS*** *is undecidable.*

Proof. Since by Lemma 19 for a given two-counter automaton M we can effectively construct a set of **P**-formulas Γ_M such that M terminates on input $(0,0)$ iff Γ_M is not consistent. It follows that **HALT** $\leq$ **CONS**. Hence, since **HALT** is undecidable, we have shown that **CONS** is undecidable too. $\square$