Inhabitation in $\lambda 2$

Florian Starke

September 14, 2015

A $\boldsymbol{\lambda 2}$ type is inhabited in $\boldsymbol{\lambda 2}$ iff there is a closed $\boldsymbol{\lambda 2}$ term of this type. The inhabitation problem in $\boldsymbol{\lambda 2}$ is to determine whether a given $\boldsymbol{\lambda 2}$ type is inhabited. This work gives a formal proof for the fact that the inhabitation problem in $\boldsymbol{\lambda 2}$ is undecidable.

1 Introduction

We will only consider the explicitly typed $\boldsymbol{\lambda 2}$ calculus (Church style), so whenever we speak of $\boldsymbol{\lambda 2}$ terms we know that the type information is given explicitly. Let us take a look at the problem. It is clear that there are closed $\boldsymbol{\lambda 2}$ terms to which no $\boldsymbol{\lambda 2}$ type can be assigned (e.g. to the $\boldsymbol{\lambda 2}$ term $(\lambda x : \forall\alpha\alpha.xx)(\lambda x : \forall\alpha\alpha.xx)$ no type can be assigned). But there are also $\boldsymbol{\lambda 2}$ types which can not be assigned to any closed $\boldsymbol{\lambda 2}$ term. We say that these types are empty. For instance the $\boldsymbol{\lambda 2}$ type $\forall\alpha\alpha$ is empty.

In what follows we will prove that the inhabitation problem in $\boldsymbol{\lambda 2}$ is undecidable. We do this by reducing the halting problem for two-counter automaton to the consistency problem of System $\mathbf{P}$ (a restricted version of first-order logic). Finally we reduce the consistency problem to the inhabitation problem in $\boldsymbol{\lambda 2}$. The constructions used for this are mainly based on [2] but the proofs go much more into detail.

2 Basic Definitions

2.1 Conventions

For variable names we will use the following conventions.
$\boldsymbol{\lambda 2}$ types: $t, t', t'', t_1, t_2, \dots, s, s_1, s_2, \dots$
$\boldsymbol{\lambda 2}$ terms: $M, M', M_1, M_2, \dots, N, N', N_1, N_2, \dots$
first-order terms: $t, t_1, t_2, \dots$
first-order formulas: $\varphi, \varphi_1, \varphi_2, \psi, \psi'$
type-variables: $p, p_1, p_2, \alpha, a, \alpha_1, a_1, \alpha_2, a_2, \dots, \beta, b, \beta_1, b_1, \beta_2, b_2, \dots$
value-variables: $x, y, z, x_1, x_2, \dots$
predicate-symbols: $P, P^1, P^2, \dots$
$\mathbf{P}$-variables: $\alpha, a, \alpha_1, a_1, \alpha_2, a_2, \dots, \beta, b, \beta_1, b_1, \beta_2, b_2, \dots$

P-formulas: $A, A', B, B', A_1, A_2, \dots$
states: $Q, Q', \widehat{Q}, Q_f, Q_0, Q_1, Q_2, \dots$
If possible we will use Greek letters for bound type-variables and Latin letters for free type-variables.

2.2 $\boldsymbol{\lambda}$-calculus $\boldsymbol{\lambda 2}$

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

Definition 1. The set of all $\boldsymbol{\lambda 2}$ 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 $\forall \alpha.t \in \mathrm{T}$.

The set of all $\boldsymbol{\lambda 2}$ 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:

- $\mathcal{V}_V \subseteq \Lambda_{\mathrm{T}}$,
- if $M_1, M_2 \in \Lambda_{\mathrm{T}}$ then $M_1 M_2 \in \Lambda_{\mathrm{T}}$,
- if $x \in \mathcal{V}_V$, $t \in \mathrm{T}_{\lambda 2}$, and $M \in \Lambda_{\mathrm{T}}$ then $\lambda x : t.M \in \Lambda_{\mathrm{T}}$,
- if $\alpha \in \mathcal{V}_T$ and $M \in \Lambda_{\mathrm{T}}$ then $\Lambda \alpha.M \in \Lambda_{\mathrm{T}}$, and
- if $M \in \Lambda_{\mathrm{T}}$ and $t \in \mathrm{T}_{\lambda 2}$ then $M\,t \in \Lambda_{\mathrm{T}}$.

If we have a type of the form $(t_1 \to (t_2 \to (\cdots \to (t_{n-1} \to t_n) \cdots)))$ we will often omit the brackets and just write $(t_1 \to t_2 \to \cdots \to t_{n-1} \to t_n)$ or $t_1 \to t_2 \to \cdots \to t_{n-1} \to t_n$ instead.

Definition 2. Let $M, N \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and $x \in \mathcal{V}_V$. The substitution of x by N in M, denoted by $M\,[x := N]$ is defined as follows:

$$M\,[x := N] = \begin{cases} N & \text{if } M = x \\ y & \text{if } M = y \text{ and } y \neq x \\ (M_1\,[x := N])(M_2\,[x := N]) & \text{if } M = M_1 M_2 \\ \lambda x : t.M' & \text{if } M = \lambda x : t.M' \\ \lambda y : t.(M'\,[x := N]) & \text{if } M = \lambda y : t.M' \text{ and } y \neq x \\ \Lambda \alpha.(M'\,[x := N]) & \text{if } M = \Lambda \alpha.M' \\ (M'\,[x := N])\,t & \text{if } M = M'\,t \end{cases}$$

Let $t, t' \in \mathrm{T}_{\lambda 2}$ and $a \in \mathcal{V}_T$. The substitution of a by t in t', denoted by $t\,[a := t']$ is defined as follows:

$$t\left[a := t'\right] = \begin{cases} t' & \text{if } t = a \\ b & \text{if } t = b \text{ and } b \neq a \\ (t_1\,[a := t']) \to (t_2\,[a := t']) & \text{if } t = t_1 \to t_2 \\ \forall a.t'' & \text{if } t = \forall a.t'' \\ \forall \beta.(t''\,[a := t']) & \text{if } t = \forall \beta.t'' \text{ and } \beta \neq a \end{cases}$$

Let $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$, $a \in \mathcal{V}_T$, and $t \in \mathrm{T}_{\lambda 2}$. The substitution of a by t in M, denoted by $M\,[a := t]$ is defined as follows:

$$M\,[a := t] = \begin{cases} x & \text{if } M = x \\ (M_1\,[a := t])(M_2\,[a := t]) & \text{if } M = M_1 M_2 \\ \lambda x : t'\,[a := t]\,.(M'\,[a := t]) & \text{if } M = \lambda x : t'.M' \\ M & \text{if } M = \Lambda a.M' \\ \Lambda \beta.(M'\,[a := t]) & \text{if } M = \Lambda \beta.M' \text{ and } \beta \neq a \\ (M'\,[a := t])\,t\,[a := t] & \text{if } M = M'\,t \end{cases}$$

In the following we will often abbreviate $(\ldots(M\,[a_n := b_n])\ldots)\,[a_1 := b_1]$ to $M\left[\vec{a} := \vec{b}\right]$ where $\vec{a} = a_1 \ldots a_n$ and $\vec{b} = b_1 \ldots b_n$.

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

$$\mathrm{FV}(M) = \begin{cases} \{x\} & \text{if } M = x \\ \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) & \text{if } M = M_1 M_2 \\ \mathrm{FV}(M') \setminus \{x\} & \text{if } M = \lambda x : t.M' \\ \mathrm{FV}(M') & \text{if } M = \Lambda \alpha.M' \\ \mathrm{FV}(M') & \text{if } M = M'\,t \end{cases}$$

The set of bound variables of M, denoted by $\mathrm{BV}(M)$, is defined as follows:

$$\mathrm{BV}(M) = \begin{cases} \emptyset & \text{if } M = x \\ \mathrm{BV}(M_1) \cup \mathrm{BV}(M_2) & \text{if } M = M_1 M_2 \\ \mathrm{BV}(M') \cup \{x\} & \text{if } M = \lambda x : t.M' \\ \mathrm{BV}(M') & \text{if } M = \Lambda \alpha.M' \\ \mathrm{BV}(M') & \text{if } M = M'\,t \end{cases}$$

Definition 4. Let $t \in \mathrm{T}_{\lambda 2}$. The set of free type-variables of t, denoted by $\mathrm{FV}(t)$, is defined inductively as follows:

$$\mathrm{FV}(t) = \begin{cases} \{a\} & \text{if } t = a \\ \mathrm{FV}(t_1) \cup \mathrm{FV}(t_2) & \text{if } t = t_1 \to t_2 \\ \mathrm{FV}(t') \setminus \{\alpha\} & \text{if } t = \forall \alpha.t' \end{cases}$$

The set of bound type-variables of t, denoted by $\mathrm{BV}(t)$, is defined inductively as follows:

$$\mathrm{BV}(t) = \begin{cases} \emptyset & \text{if } t = a \\ \mathrm{BV}(t_1) \cup \mathrm{BV}(t_2) & \text{if } t = t_1 \to t_2 \\ \mathrm{BV}(t') \cup \{\alpha\} & \text{if } t = \forall \alpha . t' \end{cases}$$

Now we can lift this definition to terms.

Definition 5. Let $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$. The set of free type-variables of M, denoted by $\mathrm{FTV}(M)$, is the union of all sets of free type-variables of types occurring in M.

The set of bound type-variables of M, denoted by $\mathrm{BTV}(M)$, is the union of all sets of bound type-variables of types occurring in M.

Definition 6. The β-reduction, denoted by $\to_\beta$, is a binary relation on $\Lambda_{\mathrm{T}_{\lambda 2}}$. For all $M, N \in \Lambda_{\mathrm{T}_{\lambda 2}}$, $x \in \mathcal{V}_V$, $t \in \mathrm{T}_{\lambda 2}$, and $\alpha \in \mathcal{V}_T$ if $\mathrm{BV}(M) \cap \mathrm{FV}(N) = \emptyset$ then $(\lambda x : t.M)N \to_\beta M\,[x := N]$ and from $\mathrm{BTV}(M) \cap \mathrm{FTV}(N) = \emptyset$ it follows that $(\Lambda \alpha . M)t \to_\beta M\,[\alpha := t]$.

The α_1-conversion, denoted by $\to_{\alpha_1}$, is a binary relation on $\Lambda_{\mathrm{T}_{\lambda 2}}$. For all $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$, $x, x' \in \mathcal{V}_V$, $t \in \mathrm{T}_{\lambda 2}$, and $\alpha, \beta \in \mathcal{V}_T$ if $x' \notin \mathrm{FV}(M) \cup \mathrm{BV}(M)$ then $\lambda x : t.M \to_{\alpha_1} \lambda x' : t.(M\,[x := x'])$ and from $\beta \notin \mathrm{FTV}(M) \cup \mathrm{BTV}(M)$ it follows that $\Lambda \alpha . M \to_{\alpha_1} \Lambda \beta .(M\,[\alpha := \beta])$.

The α_2-conversion, denoted by $\to_{\alpha_2}$, is a binary relation on $\mathrm{T}_{\lambda 2}$. For all $t \in \mathrm{T}_{\lambda 2}$, and $\alpha, \beta \in \mathcal{V}_T$ if $\beta \notin \mathrm{FV}(t) \cup \mathrm{BV}(t)$ then $\forall \alpha . t \to_{\alpha_2} \forall \beta .(t\,[\alpha := \beta])$.

Note that right now we are not able to reduce terms within a context (e.g there is no $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $\lambda x : t.(\lambda y : t.y)x \to_\beta M$).

Definition 7. So, for a binary relation $\to$ on $\Lambda_{\mathrm{T}_{\lambda 2}}$ we define the closure of $\to$ under term contexts, denoted by $\mathrm{cl}_\Lambda(\to)$, as the smallest binary relation $\Rightarrow$ on $\Lambda_{\mathrm{T}_{\lambda 2}}$ containing $\to$ such that for all $N, M, M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$, $x \in \mathcal{V}_V$, $t \in \mathrm{T}_{\lambda 2}$, and $\alpha \in \mathcal{V}_T$. If $M \Rightarrow M'$ then

$$MN \Rightarrow M'N \qquad \lambda x : t.M \Rightarrow \lambda x : t.M' \qquad M\,t \Rightarrow M'\,t$$
$$NM \Rightarrow NM' \qquad \Lambda \alpha . M \Rightarrow \Lambda \alpha . M'$$

also hold.

For a binary relation $\to'$ on $\mathrm{T}_{\lambda 2}$ we define the closure of $\to'$ under type contexts, denoted by $\mathrm{cl}_{\mathrm{T}}(\to')$, as the smallest binary relation $\Rightarrow$ on $\mathrm{T}_{\lambda 2}$ containing $\to'$ such that for all $s, t, t' \in \mathrm{T}_{\lambda 2}$, and $\alpha \in \mathcal{V}_T$. If $t \Rightarrow t'$ then

$$\forall \alpha . t \Rightarrow \forall \alpha . t' \qquad t \to s \Rightarrow t' \to s \qquad s \to t \Rightarrow s \to t'$$

also hold.

Definition 8.

We define $\Rightarrow_\beta$ as $\mathrm{cl}_\Lambda(\to_\beta)$.

And we define $\Rightarrow_\alpha$ as union of $\mathrm{cl}_\Lambda(\to_{\alpha_1})$ and $\mathrm{cl}_\Lambda(\Rightarrow_{\alpha_2})$ where

$$\Rightarrow_{\alpha_2} := \{(M\,t, M\,t'), (\lambda x : t.M, \lambda x : t'.M) \mid M \in \Lambda_{\mathrm{T}_{\lambda 2}}, x \in \mathcal{V}_V, (t, t') \in \mathrm{cl}_{\mathrm{T}}(\to_{\alpha_2})\}.$$

Finally we define $\Rightarrow_\lambda$ as $\Rightarrow_\alpha^* \circ \Rightarrow_\beta$.

Definition 9. Let $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$. The term M is in normal form if there is no $N \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $M \Rightarrow_\lambda N$.

M is weakly normalizing if there exists an $N \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that N is in normal form and $M \Rightarrow_\lambda N$.

The term M is called strongly normalizing if there is no infinite chain $M \Rightarrow_\lambda M_1 \Rightarrow_\lambda M_2 \ldots$.

Definition 10. Let $\mathcal{V} = \{x_1, \ldots, x_n\}$ be a finite subset of $\mathcal{V}_V$ such that $x_i \neq x_j$ for $1 \leq i < j \leq n$ and $t_1, \ldots, t_n \in \mathrm{T}_{\lambda 2}$. A $\boldsymbol{\lambda 2}$-basis $\Gamma = \{(x_1 : t_1), \ldots, (x_n : t_n)\}$ 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 Γ, denoted by $\mathrm{FV}(\Gamma)$, are $\bigcup\{\mathrm{FV}(t) \mid (x : t) \in \Gamma\}$.

For a basis Γ and another basis Σ such that $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Sigma) = \emptyset$, $x \in \mathcal{V}_V \setminus \mathrm{dom}(\Gamma)$, and $t \in \mathrm{T}_{\lambda 2}$ we will abbreviate $\Gamma \cup \{(x : t)\}$ to $\Gamma, x : t$ and $\Gamma \cup \Sigma$ to Γ, Σ.

Definition 11. Let M be in $\Lambda_{\mathrm{T}_{\lambda 2}}$, t in $\mathrm{T}_{\lambda 2}$, and Γ be a basis. A statement $M : t$ is derivable from Γ, denoted by $\Gamma \vdash M : t$, if $M : t$ can be produced using the following rules.

(Axiom)	$\Gamma, x : t \vdash x : t$	
(λ-Introduction)	$\dfrac{\Gamma, x : t_1 \vdash M : t_2}{\Gamma \vdash \lambda x : t_1.M : t_1 \to t_2}$	
(λ-Elimination)	$\dfrac{\Gamma \vdash M_1 : t_1 \to t_2 \quad \Gamma \vdash M_2 : t_1}{\Gamma \vdash M_1 M_2 : t_2}$	
($\forall$-Introduction)	$\dfrac{\Gamma \vdash M : t}{\Gamma \vdash \Lambda\alpha.M : \forall\alpha.t}$	$\alpha \notin \mathrm{FV}(\Gamma)$
($\forall$-Elimination)	$\dfrac{\Gamma \vdash M : \forall\alpha.t}{\Gamma \vdash M\,t' : t\,[\alpha := t']}$	

Definition 12. A term $M \in \Lambda_{\mathrm{T}_{\lambda 2}}$ is well typed if there exists a basis Γ and a type $t \in \mathrm{T}_{\lambda 2}$ such that $\Gamma \vdash M : t$ holds.

The following two theorems are well known (for formal proofs see [1]).

Theorem 13. *Let M, M' be in $\Lambda_{T_{\lambda 2}}$ and $M \Rightarrow_\alpha^* M'$ or $M \Rightarrow_\beta^* M'$, t in $T_{\lambda 2}$, and Γ be a basis. If $\Gamma \vdash M : t$ then $\Gamma \vdash M' : t$.*

Theorem 14. *All well typed $\boldsymbol{\lambda 2}$ terms are strongly normalizing.*

Definition 15. The inhabitation problem for $\boldsymbol{\lambda 2}$, denoted by **INHAB**, is defined as follows. Given a $\boldsymbol{\lambda 2}$ type t.

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

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

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

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

2.3 First-order logic

Definition 16. A ranked set is a tuple (Σ, rk), where Σ is a countable set and $rk\colon \Sigma \to \mathbb{N}$ 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 in Σ with a certain rank k, denoted by $\Sigma^{(k)}$, is defined as $\Sigma^{(k)} := rk^{-1}(k)$.

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

Definition 17. 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 $\to$ on formulas, where for some φ, $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$ the formula $(\varphi \to \psi)$ is defined as $(\neg\varphi \vee \psi)$, if we have a formula of the form $(\varphi_1 \to (\varphi_2 \to (\cdots \to (\varphi_{n-1} \to \varphi_n)\cdots)))$ we will often omit the brackets and just write $(\varphi_1 \to \varphi_2 \to \cdots \to \varphi_{n-1} \to \varphi_n)$ or $\varphi_1 \to \varphi_2 \to \cdots \to \varphi_{n-1} \to \varphi_n$ instead.

For nullary relation symbols P we will abbreviate $P()$ to P. If a formula φ is of the form $Qy.\psi$ (where $Q \in \{\exists, \forall\}$, $y \in \mathcal{V}$, and $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$) we often drop the dot and write $Qy\psi$ instead. If a formula φ has multiple variables bound by the same quantifier (i.e. $\varphi = Qy_1.Qy_2.\ldots Qy_n.\psi$ for $Q \in \{\exists, \forall\}$, some $n \in \mathbb{N}$, $y_1, y_2, \ldots, y_n \in \mathcal{V}$, and $\psi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$) we abbreviate φ to $Qy_1y_2\ldots y_n.\psi$ or to $Q\vec{y}.\psi$ where $\vec{y} = y_1y_2\ldots y_n$.

Definition 18. The set of variables of a term $t \in \mathcal{T}_{(\mathcal{V},\mathcal{F})}$, denoted by $\mathrm{V}(t)$, is 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 set of free variables of a formula $\varphi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$, denoted by $\mathrm{FV}(\varphi)$, is 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 19. The set of subformulas of a formula $\varphi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$, denoted by $\mathrm{SUB}(\varphi)$, is defined as follows:

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

Definition 20. We say that a formula $\varphi \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$ contains no dummy quantifiers if for all $\psi \in \mathrm{SUB}(\varphi)$ of the form $\psi = \forall y.\psi'$ or $\psi = \exists y.\psi'$ for some $y \in \mathcal{V}$ and some $\psi' \in \mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$ we have that $y \in \mathrm{FV}(\psi')$.

Definition 21. 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 $t\,[y := t']$, 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 φ be in $\mathcal{L}_{(\mathcal{V},\mathcal{F},\mathcal{P})}$. The substitution of y by t' in φ, denoted by $\varphi\,[y := t']$, is defined as follows:

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

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

Definition 22. An interpretation I over $\mathcal{V}$, $\mathcal{F}$, and $\mathcal{P}$ is a triple $I = (\Delta, \cdot^I, \omega)$, where

Δ	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 $k \in \mathbb{N}$, $P \in \mathcal{P}^{(k)}$
ω	is a function from $\mathcal{V}$ to Δ.

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 23. 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 φ be a formula. The interpretation of φ under I, denoted by φ^I, is defined recursively as follows:

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

The interpretation I is a model of φ, 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 24. Let Γ be a finite set of first-order formulas.

We say that an interpretation I is a model of Γ, denoted by $I \models \Gamma$, 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 $\bigcup\{\mathrm{FV}(\varphi) \mid \varphi \in \Gamma\}$.

2.4 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 25. 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 $C = \langle Q, m, n\rangle$, where $Q \in \mathcal{Q}$ and $m, n \in \mathbb{N}$. Let r be in $R(\mathcal{Q} \setminus \{Q_f\})$, then $\Rightarrow^r_M$ 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 $\bigcup_{r \in R(Q \setminus \{Q_f\})} \Rightarrow^r_M$.

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 $D_1, \ldots, D_i$ 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 $i - 1$).

Definition 26. 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 $\mathcal{V}_P = \{\alpha, a, \beta, b, \ldots\}$ be a countably infinite subset of $\mathcal{V}_T$ (of variables). Let $\mathcal{P}_P = \{P, Q, \ldots\}$ be a set (of predicate symbols) and $\mathcal{P}$ a ranked set such that $\mathcal{P}^{(0)} = \{\textbf{false}\}$, $\mathcal{P}^{(2)} = \mathcal{P}_P$, and $\mathcal{P}^{(k)} = \emptyset$ for all $k \in \mathbb{N} \setminus \{0, 2\}$. A first-order logic formula φ over $\mathcal{V}_P$, $\emptyset$, and $\mathcal{P}$ 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 $\vec{\alpha} = \alpha_1 \dots \alpha_m$ where $\alpha_1, \dots, \alpha_m \in \mathcal{V}_P$, 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 $\alpha \in \mathrm{FV}(A_n) \cap \mathrm{BV}(\varphi)$ there exists an $i \in \{1, \dots, n-1\}$ such that $\alpha \in \mathrm{FV}(A_i)$.

existential formula if there is a $\vec{\alpha} = \alpha_1 \dots \alpha_m$ where $\alpha_1, \dots, \alpha_m \in \mathcal{V}_P$, an $n \in \mathbb{N}^+$, atomic formulas $A_i \neq \mathbf{false}$ for $i \in \{1, \dots, n\}$, $\beta \in \mathcal{V}_P$, such that for each $\alpha \in (\mathrm{FV}(A_n) \cap \mathrm{BV}(\varphi)) \setminus \{\beta\}$ there exists an $i \in \{1, \dots, n-1\}$ such that $\alpha \in \mathrm{FV}(A_i)$ and $\varphi = \forall\vec{\alpha}(A_1 \to A_2 \to \cdots \to A_{n-1} \to \forall\beta(A_n \to \mathbf{false}) \to \mathbf{false})$.

The set of formulas of System **P** (= set of **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. In what follows we assume all **P**-formulas to contain no dummy quantifiers.

Definition 27. A finite set of **P**-formulas Γ is called **P**-basis, or basis if it is clear from the context whether a **P**-basis or a $\boldsymbol{\lambda}$**2**-basis is meant.

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

Definition 28. Let A be a **P**-formula, and Γ be a basis. The formula A is a semantic consequence of Γ, denoted by $\Gamma \vdash A$, if A can be produced using the following 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)	$\dfrac{\Gamma \vdash \forall\alpha B}{\Gamma \vdash B\,[\alpha := b]}$	$b \in \mathcal{V}_P$

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

Definition 29. Let Γ be a basis. The **P**-formula A is a semantic consequence with falsity of Γ, denoted by $\Gamma \vdash_{\mathrm{f}} A$, if for every interpretation I

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

This allows us to add the following deduction rule.

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

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

$$
\begin{aligned}
I \models \Gamma, \forall\alpha(A \to \mathbf{false}) \to \mathbf{false} &\Rightarrow I \models \forall\alpha(A \to \mathbf{false}) \to \mathbf{false} \\
&\Rightarrow \text{not } (\forall\alpha(A \to \mathbf{false}))^I \text{ or } \mathbf{false}^I \\
&\Rightarrow \text{not } (\forall\alpha(A \to \mathbf{false}))^I \text{ or } \bot \\
&\Rightarrow \text{not } (\forall\alpha(A \to \mathbf{false}))^I \\
&\Rightarrow \text{not (forall } d \in \Delta\colon (A \to \mathbf{false})^{I[\alpha \mapsto d]}) \\
&\Rightarrow \text{exists } d \in \Delta\colon \text{not (not } A^{I[\alpha \mapsto d]} \text{ or } \mathbf{false}^{I[\alpha \mapsto d]}) \\
&\Rightarrow \text{exists } d \in \Delta\colon \text{not (not } A^{I[\alpha \mapsto d]} \text{ or } \bot) \\
&\Rightarrow \text{exists } d \in \Delta\colon \text{not (not } A^{I[\alpha \mapsto d]}) \\
&\Rightarrow \text{exists } d \in \Delta\colon A^{I[\alpha \mapsto d]}
\end{aligned}
$$

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. $\square$

Definition 30. The consistency problem, denoted by **CONS**, is defined as follows. Given a set of **P**-formulas Γ.

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

3.2 Consistency in System P 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 Γ_M such that

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

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

The intended informal meaning for these new relation symbols is the following:

- The meaning of $Q(a)$ is "a represents a configuration and Q is the state of this configuration".

- For $i \in \{1,2\}$, $R_i(a,m)$ denotes that "the value of register i in the configuration represented by a is represented by m" (we call m anchor of a for register i).
- With $S(a,b)$ we state that "b is a successor of a".
- The meaning of $P(a,b)$ is "b is a predecessor of a".
- And $E(a)$ marks "a as the end of chain".
- With $D(a)$ we state that "a is not the end of a chain".
- Finally $G(a)$ has no actual meaning, it holds for all elements representing a configuration or a number. But we just need it for the existential formulas.

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

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

Next we need sets of **P**-formulas for all possible transitions. For every $Q \in \mathcal{Q} \setminus \{Q_f\}$ and $r \in \mathcal{R}_{\mathcal{Q}}$ 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
- $\forall \alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_2(\alpha,\gamma) \to R_2(\beta,\gamma))$
  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 if possible

- $\forall \alpha\beta\gamma(Q(\alpha) \to S(\alpha, \beta) \to R_2(\alpha, \gamma) \to R_2(\beta, \gamma))$
  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 Γ_1 to ensure that our representation works correctly. The following formula are in Γ_1:

- $\forall \alpha\beta(S(\alpha, \beta) \to G(\beta))$
- $\forall \alpha(D(\alpha) \to G(\alpha))$
- $\forall \alpha\beta(P(\alpha, \beta) \to D(\alpha))$
  no element with a predecessor is the end of a chain
- $\forall \alpha(G(\alpha) \to \forall \beta(R_1(\alpha, \beta) \to \mathbf{false}) \to \mathbf{false})$
  every element that represents a configuration has a value for register 1
- $\forall \alpha(G(\alpha) \to \forall \beta(R_2(\alpha, \beta) \to \mathbf{false}) \to \mathbf{false})$
  every element that represents a configuration has a value for register 2
- $\forall \alpha(G(\alpha) \to \forall \beta(S(\alpha, \beta) \to \mathbf{false}) \to \mathbf{false})$
  every element that represents a configuration has a successor

Note that in the last three formulas the only task of $G(\alpha)$ is to make these formulas existential formulas (e.g. $\forall \alpha(\forall \beta(S(\alpha, \beta) \to \mathbf{false}) \to \mathbf{false})$ is not an existential formula).

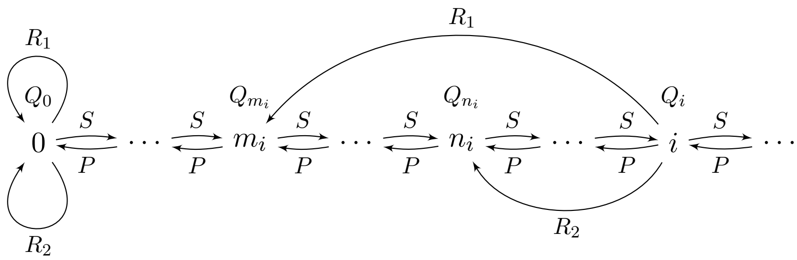
We define $\Gamma_{\overline{M}}$ as $\bigcup_{Q \in \mathcal{Q} \setminus \{Q_f\}} \Gamma_{Q,R(Q)} \cup \{\forall \alpha(Q_f(\alpha) \to \mathbf{false})\} \cup \Gamma_1$. We have added the formula $\forall \alpha(Q_f(\alpha) \to \mathbf{false})$ to be able to deduce **false** if our automaton terminates. Finally we can define Γ_M as $\Gamma_{C_0} \cup \Gamma_{\overline{M}}$, where $C_0 = \langle Q_0, 0, 0\rangle$ is the initial configuration.

Lemma 31.

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

Proof. Assume M does not terminate it follows that there is an infinite chain $C_0 \Rightarrow_M C_1 \Rightarrow_M C_2 \Rightarrow_M \cdots$ ($C_i = \langle Q_i, m_i, n_i\rangle$ for $i \in \mathbb{N}$). 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 $d_1 \xrightarrow{R} d_2$ we say that $(d_1, d_2) \in R^I$. And we use $\overset{P}{d}$ 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 will also represent the i^{th} configuration of our infinite computation. Now the idea for our model of Γ_M looks like this:



We have $0 \in E^I$, all other numbers are in D^I, and all numbers are in G^I.
Here is the more formal definition of our model $I = (\mathbb{N}, \cdot^I, \omega)$.

$$
\begin{array}{lll}
P^I = \{(i+1, i) \mid i \in \mathbb{N}\} & R_1^I = \{(i, m_i) \mid i \in \mathbb{N}\} & R_2^I = \{(i, n_i) \mid i \in \mathbb{N}\} \\
S^I = \{(i, i+1) \mid i \in \mathbb{N}\} & D^I = \{(i, i) \mid i \in \mathbb{N}^+\} & E^I = \{(0, 0)\} \\
Q^I = \{(i, i) \mid i \in \mathbb{N}, Q = Q_i\} \text{ for every } Q \in \mathcal{Q} & & \mathbf{false}^I = \bot \\
G^I = \mathbb{N} & &
\end{array}
$$

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

Since there are no free variables in Γ_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 Γ_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 (this is why we defined $\Gamma_{\overline{M}}$ and not just Γ_M right away).

Lemma 32. *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 $\widehat{m}, \widehat{n} \in \mathbb{N}$) 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 $m, n \in \mathbb{N}$. Hence, $Q_f(a)$ is in Γ_C for some $a \in \mathcal{V}_P$ and $\forall\alpha(Q_f(\alpha) \to \mathbf{false})$ is in $\Gamma_{\overline{M}}$, we can easily deduce **false**.

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

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 $\bot$ (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 $C \Rightarrow^r_M D$ 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 Γ_C contains the formulas:

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

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

$Q(a)$, $E(a_m)$, $E(b_n)$, $G(a)$, $G(a_m)$, and $G(b_n)$.

And Γ_D contains the formulas:

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

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

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

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

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

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 $r = -(1, Q_1, Q_2)$, because 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.

In every case 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. In the following 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 the new configuration D (let $b \in \mathcal{V}_P \setminus \mathrm{FV}(\Gamma_C)$, note that $\mathrm{FV}(\Gamma_{\overline{M}}) = \emptyset$).

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

For the new variable b we have to deduce $G(b)$. Again we will just drop $S(a, b)$ on the left side for comprehensibility.

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

Since register 2 should not change we need $R_2(b, b_0)$.

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

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 the configuration represented by b is Q'.

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

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

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

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

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

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

$$
\dfrac{\dfrac{\dfrac{\vdots}{D(d) \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} D(d) \to \mathbf{f}} \qquad \dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha\beta(P(\alpha,\beta) \to D(\alpha))}{\vdash_{\mathrm{f}} P(d,a_0) \to D(d)} \quad \vdash_{\mathrm{f}} P(d,a_0)}{\vdash_{\mathrm{f}} D(d)}}{\vdash_{\mathrm{f}} \mathbf{f}}
$$

Now we can easily deduce $G(d)$.

$$
\dfrac{\dfrac{\dfrac{\vdots}{G(d) \vdash_{\mathrm{f}} \mathbf{f}}}{\vdash_{\mathrm{f}} G(d) \to \mathbf{f}} \qquad \dfrac{\dfrac{\vdash_{\mathrm{f}} \forall\alpha(D(\alpha) \to G(\alpha))}{\vdash_{\mathrm{f}} D(d) \to G(d)} \quad \vdash_{\mathrm{f}} D(d)}{\vdash_{\mathrm{f}} G(d)}}{\vdash_{\mathrm{f}} \mathbf{f}}
$$

Now we have already deduced Γ_D. To see why we define $\widehat{a} := b$, $\widehat{b}_i := b_i$ for $i \in \{0, \dots, n\}$, $\widehat{a}_0 := d$, and $\widehat{a}_{i+1} := a_i$ for $i \in \{0, \dots, m\}$. It follows that $\Gamma_D \subseteq (\Gamma_C \cup \{S(a,b), G(b), Q'(b), R_2(b,b_0), R_1(b,d), P(d,a_0), D(d), G(d)\})$. 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.

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\UnaryInfC{$Q_2(b) \vdash_{\mathrm{f}} \mathbf{f}$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q_2(b) \to \mathbf{f}$}
\AxiomC{$\vdash_{\mathrm{f}} \forall\alpha\beta\gamma(Q(\alpha) \to S(\alpha,\beta) \to R_1(\alpha,\gamma) \to E(\gamma) \to Q_2(\beta))$}
\UnaryInfC{$\vdash_{\mathrm{f}} Q(a) \to S(a,b) \to R_1(a,a_0) \to E(a_0) \to Q_2(b)$}
\AxiomC{$\vdash_{\mathrm{f}} Q(a)$}
\BinaryInfC{$\vdash_{\mathrm{f}} S(a,b) \to R_1(a,a_0) \to E(a_0) \to Q_2(b)$}
\AxiomC{$\vdash_{\mathrm{f}} S(a,b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} R_1(a,a_0) \to E(a_0) \to Q_2(b)$}
\AxiomC{$\vdash_{\mathrm{f}} R_1(a,a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} E(a_0) \to Q_2(b)$}
\AxiomC{$\vdash_{\mathrm{f}} E(a_0)$}
\BinaryInfC{$\vdash_{\mathrm{f}} Q_2(b)$}
\BinaryInfC{$\vdash_{\mathrm{f}} \mathbf{f}$}
\end{prooftree}
$$

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 $i \in \{0,\dots,n\}$, and $\widehat{a}_0 := a_0$ then it is clear that we have deduced all formulas required for Γ_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.

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

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

Lemma 33.

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

Proof. The $\Leftarrow$ direction is proven in Lemma 31. And the $\Rightarrow$ direction is a direct consequence of Lemma 32 with $C = \langle Q_0, 0, 0\rangle$. $\square$

Theorem 34. *The consistency problem is undecidable.*

Proof. Since by Lemma 33 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**. Since **HALT** is undecidable we have shown that **CONS** is undecidable too. $\square$

4 Inhabitation in $\boldsymbol{\lambda}2$ is undecidable

Now we can show that the inhabitation problem in $\boldsymbol{\lambda 2}$ is undecidable by reducing **CONS** to **INHAB**. Given a **P**-basis Γ we construct a $\boldsymbol{\lambda 2}$-basis $\overline{\Gamma}$ such that

$$\Gamma \vdash \textbf{\textit{false}} \qquad \textit{iff} \qquad \textit{There is a } \boldsymbol{\lambda 2} \textit{ term } M \textit{ such that } \overline{\Gamma} \vdash M : \textbf{\textit{false}}$$

where **false** $\in \mathcal{V}_T$. Furthermore for every $P \in \mathcal{P}_P$ we have $p, p_1, p_2 \in \mathcal{V}_T$.

Definition 35. For a **P**-formula A we define the <u>code</u> of A, denoted by $\overline{A}$, as follows.
If A is an atomic formula then

$$\overline{A} = \begin{cases} \textbf{false} & \text{if } A = \textbf{false} \\ (\alpha \to p_1) \to (\beta \to p_2) \to p & \text{if } A = P(\alpha,\beta) \end{cases}$$

We will abbreviate $(\alpha \to p_1) \to (\beta \to p_2) \to p$ to $P_{\alpha\beta}$.

If A is a universal formula, it follows that there is an $n \in \mathbb{N}$, atomic formulas $A_1, A_2, \ldots, A_n$, and an $\vec{\alpha} = \overline{\alpha}_1 \ldots \overline{\alpha}_m$ for some $m \in \mathbb{N}$ and $\overline{\alpha}_1, \ldots, \overline{\alpha}_m \in \mathcal{V}_P$ such that $A = \forall\vec{\alpha}(A_1 \to A_2 \to \cdots \to A_n)$, then

$$\overline{A} = \forall\vec{\alpha}(\overline{A_1} \to \overline{A_2} \to \cdots \to \overline{A_n})$$

If A is an existential formula, it follows that for some $n \in \mathbb{N}^+$, some atomic formulas $A_1, \ldots, A_n$, some $\vec{\alpha} = \overline{\alpha}_1 \ldots \overline{\alpha}_m$ for some $m \in \mathbb{N}$ and $\overline{\alpha}_1, \ldots, \overline{\alpha}_m \in \mathcal{V}_P$, and some $\beta \in \mathcal{V}_P$ it holds that $A = \forall\vec{\alpha}(A_1 \to \cdots \to A_{n-1} \to \forall\beta((A_n) \to \mathbf{false}) \to \mathbf{false})$, then

$$\overline{A} = \forall\vec{\alpha}(\overline{A_1} \to \cdots \to \forall\beta(\overline{A_n} \to \mathbf{false}) \to \mathbf{false})$$

For a $\mathbf{P}$-basis Γ we define the code of Γ, denoted by $\overline{\Gamma}$, as $\{(x_A : \overline{A}) \mid A \in \Gamma\}$.

In the following lemma we prove the $\Rightarrow$ direction by constructing a $\boldsymbol{\lambda 2}$ term M with the required type.

Lemma 36. *Let Γ be a $\boldsymbol{P}$-basis and A a $\boldsymbol{P}$-formula such that $\Gamma \vdash A$. Then there exists a term $M \in \Lambda_{T_{\lambda 2}}$ such that $\overline{\Gamma} \vdash M : \overline{A}$ holds.*

Proof. We proof this by induction on the structure of the proof.

A is produced by the <u>Axiom</u> rule. It follows that $A \in \Gamma$ and therefore $(x_A : \overline{A}) \in \overline{\Gamma}$. Now the term $M := x_A$ fulfills the condition.

A is produced by the <u>$\to$-Introduction</u> rule. It follows that $A = A' \to B'$ for some $\mathbf{P}$-formulas A' and B'. We can now apply the induction hypothesis to $\Gamma, A' \vdash B'$ and we get that there exists an $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $\overline{\Gamma}, \overline{A'} \vdash M' : \overline{B'}$. With the λ-Introduction rule we deduce $\overline{\Gamma} \vdash \lambda x_{A'} : \overline{A'}.M' : \overline{A'} \to \overline{B'}$. Since A has to be a universal or an existential formula $\overline{A'} \to \overline{B'} = \overline{A' \to B'}$. So $M := \lambda x_{A'} : \overline{A'}.M'$ has the required type.

A is produced by the <u>$\to$-Elimination</u> rule. So there exists a $\mathbf{P}$-formula B such that $\Gamma \vdash B \to A$ and $\Gamma \vdash B$. Now we apply the induction hypothesis and get that there exist $M_1, M_2 \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $\overline{\Gamma} \vdash M_1 : \overline{B \to A}$ and $\overline{\Gamma} \vdash M_2 : \overline{B}$. Again we have that $\overline{B \to A} = \overline{B} \to \overline{A}$. It follows that $M := M_1 M_2$ has the type $\overline{A}$.

A is produced by the <u>$\forall$-Introduction</u> rule. It follows that $A = \forall\beta B$ for some $\beta \in \mathcal{V}_P \setminus \mathrm{FV}(\Gamma)$ and some $\mathbf{P}$-formula B. By applying the induction hypothesis to $\Gamma \vdash B$ we get that there exists an $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $\overline{\Gamma} \vdash M' : \overline{B}$. We deduce that $M := \Lambda\beta.M'$ has type $\forall\beta.\overline{B} = \overline{\forall\beta B}$ as desired.

A is produced by the <u>$\forall$-Elimination</u> rule. Then there is a $\mathbf{P}$-formula B and variables $\alpha, b \in \mathcal{V}_P$ such that $\Gamma \vdash \forall\alpha B$ and $A = B\,[\alpha := b]$. The induction hypothesis implies that there exits an $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that $\overline{\Gamma} \vdash M' : \overline{\forall\alpha B}$. Since $\overline{\forall\alpha B} = \forall\alpha.\overline{B}$ the term $M := M'\,b$ has the type $\overline{B}\,[\alpha := b] = \overline{B\,[\alpha := b]}$.

$\square$

In the next two lemmas we will prove the $\Leftarrow$ direction.

Lemma 37. *Let* Γ *be a* $\boldsymbol{P}$*-basis,* $M \in \Lambda_{T_{\lambda 2}}$, $P \in \mathcal{P}_P$, *and* $s, t \in T_{\lambda 2}$ *such that* $\overline{\Gamma} \vdash M : P_{st}$ *holds. Then* $s, t \in \mathcal{V}_P$ *(remember that* $\mathcal{V}_P \subseteq \mathcal{V}_T$*). Furthermore* $\Gamma \vdash P(s,t)$ *holds.*

Proof. Note that M is a well types $\boldsymbol{\lambda 2}$ term and hence, by Theorem 13, there is a $N \in \Lambda_{\mathrm{T}_{\lambda 2}}$ such that N is in normal form and $M \Rightarrow_{\lambda}^{*} N$. From Theorem 14 it follows that the statement $N : P_{st}$ is derivable from $\overline{\Gamma}$. Therefore we can assume w.l.o.g. that M is in normal form.

We now proof the lemma by structural induction on the term M.

$\underline{M = x}$ for some $x \in \mathcal{V}_V$.
It follows that $(x : P_{st}) \in \overline{\Gamma}$. Now the definition of $\overline{\Gamma}$ yields that $P(s,t) \in \Gamma$. Therefore $s, t \in \mathcal{V}_P$ and $\Gamma \vdash P(s,t)$ holds trivially.

$\underline{M = M_1 M_2}$ for some $M_1, M_2 \in \Lambda_{\mathrm{T}_{\lambda 2}}$.
Since M is in normal form we have that $M_1 = x N_1 \dots N_k$ for some $x \in \mathcal{V}_V$, $k \in \mathbb{N}$, and some $N_1, \dots, N_k \in \Lambda_{\mathrm{T}_{\lambda 2}} \cup \mathrm{T}_{\lambda 2}$.

We conclude that $x = x_A$ and $(x : \overline{A}) \in \overline{\Gamma}$ for some universal formula $A = \forall \vec{\alpha} (P^1(\alpha_1, \beta_1) \to \cdots \to P^n(\alpha_n, \beta_n) \to P(\alpha, \beta))$ in Γ where $\vec{\alpha} = \overline{\alpha}_1 \dots \overline{\alpha}_m$ for some $m \in \mathbb{N}$ and some $\overline{\alpha}_1, \dots, \overline{\alpha}_m \in \mathcal{V}_P$.

Furthermore $M = x \vec{t} \vec{N}$ for some $\vec{t} = \overline{t}_1 \dots \overline{t}_m$ with $\overline{t}_1, \dots, \overline{t}_m \in \mathrm{T}_{\lambda 2}$ and some $\vec{N} = N_1 \dots N_n$ with $N_1, \dots, N_n \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and $\overline{\Gamma} \vdash N_i : P^i_{s_i t_i}$ (where $s_i = \alpha_i \left[\vec{\alpha} := \vec{t}\,\right]$ and $t_i = \beta_i \left[\vec{\alpha} := \vec{t}\,\right]$) for $i \in \{1, \dots, n\}$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overline{\Gamma} \vdash x : \forall \vec{\alpha} (P^1_{\alpha_1 \beta_1} \to \cdots \to P^n_{\alpha_n \beta_n} \to P_{\alpha \beta})}{\overline{\Gamma} \vdash x \vec{t} : P^1_{s_1 t_1} \to \cdots \to P^n_{s_n t_n} \to P_{st}} \qquad \overline{\Gamma} \vdash N_1 : P^1_{s_1 t_1}}{\vdots}}{\overline{\Gamma} \vdash x \vec{t} N_1 \dots N_{n-1} : P^n_{s_n t_n} \to P_{st}} \qquad \overline{\Gamma} \vdash N_n : P^n_{s_n t_n}}{\overline{\Gamma} \vdash (x \vec{t} N_1 \dots N_{n-1}) N_n : P_{st}}
$$

For $i \in \{1, \dots, n\}$ we can now apply the induction hypothesis to $\overline{\Gamma} \vdash N_i : P^i_{s_i t_i}$ and we get that $s_i, t_i \in \mathcal{V}_P$ and that $\Gamma \vdash P^i(s_i, t_i)$ holds.

If $\alpha = \overline{\alpha}_j$ for some $j \in \{1, \dots, n\}$ then because there are no dummy quantifiers we get that $s = \overline{t}_j$. Furthermore since $\alpha \in \mathrm{FV}(P(\alpha, \beta)) \setminus \mathrm{FV}(A)$ it follows that there exists an $i \in \{1, \dots, n\}$ such that $\alpha \in \mathrm{FV}(P^i(\alpha_i, \beta_i))$, i.e. $\alpha = \alpha_i$ or $\alpha = \beta_i$. It follows that $s = s_i$ or $s = t_i$, in both cases we get that $s \in \mathcal{V}_P$.

If $\alpha \neq \overline{\alpha}_j$ for all $j \in \{1, \dots, n\}$ then $\alpha \in \mathrm{FV}(A)$ and therefore $s = \alpha$ and $s \in \mathcal{V}_P$.

For t we can make a similar argument and get that $t \in \mathcal{V}_P$.

Finally we have to show that $P(s,t)$ is a semantic consequence of Γ.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \forall\vec{\alpha}(P^1(\alpha_1,\beta_1)\to\cdots\to P^n(\alpha_n,\beta_n)\to P(\alpha,\beta))}{\Gamma \vdash P^1(s_1,t_1)\to\cdots\to P^n(s_n,t_n)\to P(s,t)} \qquad \Gamma\vdash P^1(s_1,t_1)}{\vdots}}{\Gamma \vdash P^n(s_n,t_n)\to P(a,b)} \qquad \Gamma\vdash P^n(s_n,t_n)}{\Gamma\vdash P(s,t)}$$

$\underline{M = \lambda x : t'.M'}$ for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$, some $x \in \mathcal{V}_V$, and some $t' \in \mathrm{T}_{\lambda 2}$ (w.l.o.g. $x \notin \mathrm{dom}(\Gamma)$).
It follows that $t' = s \to p_1$ and $\overline{\Gamma}, x : s \to p_1 \vdash M' : (t \to p_2) \to p$.

If $M' = yx$ for some $y \in \mathcal{V}_V$ then it has to be that $y = x_{P(s,t)}$ and $(y : (s \to p_1) \to (t \to p_2) \to p) \in \overline{\Gamma}$. It follows that $P(s,t) \in \Gamma$ and therefore $s, t \in \mathcal{V}_P$ and $\Gamma \vdash P(s,t)$.

If $M' = \lambda y : t \to p_2.zxy$ for some $y, z \in \mathcal{V}_V$ then $z = x_{P(s,t)}$ and therefore $(z : (s \to p_1) \to (t \to p_2) \to p) \in \overline{\Gamma}$. We get that $P(s,t) \in \Gamma$ and conclude that $s, t \in \mathcal{V}_P$ and $\Gamma \vdash P(s,t)$.

All other cases for M' are impossible because there are no **P**-formulas A such that $\overline{A}$ has the required type.

$\underline{M = \Lambda\gamma.M'}$ for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and some $\gamma \in \mathcal{V}_T$.
It follows that $\overline{\Gamma} \vdash M : \forall\gamma.t'$ for some $t' \in \mathrm{T}_{\lambda 2}$. But this can not be since $P_{st} = (s \to p_1) \to (t \to p_2) \to p$. Therefore M is not of the form $\Lambda\gamma.M'$ and this case is impossible.

$\underline{M = M'\,t'}$ for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and some $t' \in \mathrm{T}_{\lambda 2}$.
Since M is in normal form we have that $M' = xM_1 \dots M_n$ for some $x \in \mathcal{V}_V$, $n \in \mathbb{N}$, and some $M_1, \dots, M_n \in \Lambda_{\mathrm{T}_{\lambda 2}} \cup \mathrm{T}_{\lambda 2}$. Hence, $(x : \overline{A}) \in \overline{\Gamma}$ for some **P**-formula A and $M = M'\,t'$, we get that this case is impossible because no such A exists.

The only case where the contradiction is not obvious is when A is a universal formula and $M_1, \dots, M_n \in \mathrm{T}_{\lambda 2}$. Furthermore because there are no dummy quantifiers $n \leq 1$. So A is of the form $A = \forall\vec{\alpha}(P(\alpha,\beta))$ where $\vec{\alpha} \in \{\alpha\beta, \beta\alpha, \alpha, \beta\}$. But in every case A is not a **P**-formula since there always is a $\gamma \in \mathrm{FV}(P(\alpha,\beta)) \setminus \mathrm{FV}(A)$.

$\square$

Lemma 38. *Let Γ be a **P**-basis, $M \in \Lambda_{T_{\lambda 2}}$ such that $\overline{\Gamma} \vdash M :$ **false** holds. Then $\Gamma \vdash$ **false** holds.*

Proof. By structural induction on the term M. Again we can assume that M is in normal form.

$\underline{M = x}$ for some $y \in \mathcal{V}_V$.
It follows that $(x :$ **false**$) \in \overline{\Gamma}$. Now the definition of $\overline{\Gamma}$ yields that **false** $\in \Gamma$. Therefore $\Gamma \vdash$ **false** holds.

$\underline{M = M_1 M_2}$ for some $M_1, M_2 \in \Lambda_{\mathrm{T}_{\lambda 2}}$.
Because M is in normal form we have that $M_1 = xN_1 \dots N_k$ for some $x \in \mathcal{V}_V$, $k \in \mathbb{N}$, and some $N_1, \dots N_k \in \Lambda_{\mathrm{T}_{\lambda 2}} \cup \mathrm{T}_{\lambda 2}$. We know that $x = x_A$ for some $A \in \Gamma$.

Firstly A could be a universal formula. It follows that A is of the form $A = \forall\vec{\alpha}(P^1(\alpha_1, \beta_1) \to \dots \to P^n(\alpha_n, \beta_n) \to \mathbf{false})$ where $\vec{\alpha} = \overline{\alpha}_1 \dots \overline{\alpha}_m$ for some $m \in \mathbb{N}$ and some $\overline{\alpha}_1, \dots, \overline{\alpha}_m \in \mathcal{V}_P$. In this case $M = x\vec{t}\vec{N}$ for some $\vec{t} = \overline{t}_1 \dots \overline{t}_m$ with $\overline{t}_1, \dots, \overline{t}_m \in \mathrm{T}_{\lambda 2}$ and some $\vec{N} = N_1 \dots N_n$ with $N_1, \dots, N_n \in \Lambda_{\mathrm{T}_{\lambda 2}}$. Now $\Gamma \vdash \mathbf{false}$ can be deduced as in the previous proof.

Secondly A could be an existential formula. It follows that A is of the form $A = \forall\vec{\alpha}(P^1(\alpha_1, \beta_1) \to \dots \to \forall\beta(P^n(\alpha_n, \beta_n) \to \mathbf{false}) \to \mathbf{false})$ where $\vec{\alpha} = \overline{\alpha}_1 \dots \overline{\alpha}_m$ for some $m \in \mathbb{N}$ and some $\overline{\alpha}_1, \dots, \overline{\alpha}_m \in \mathcal{V}_P$ (w.l.o.g. $\beta \notin \mathrm{FV}(\overline{\Gamma})$ and $\beta \neq \overline{\alpha}_i$ for all $i \in \{1, \dots, m\}$). Then M has to be of the form $M = x\vec{t}\vec{N}L$ for some $\vec{t} = \overline{t}_1 \dots \overline{t}_m$ with $\overline{t}_1, \dots, \overline{t}_m \in \mathrm{T}_{\lambda 2}$, some $\vec{N} = N_1 \dots N_{n-1}$ with $N_1, \dots, N_{n-1} \in \Lambda_{\mathrm{T}_{\lambda 2}}$, and some $L \in \Lambda_{\mathrm{T}_{\lambda 2}}$. It also has to hold that $\overline{\Gamma} \vdash L : \forall\beta(P^n_{s_n t_n} \to \mathbf{false})$ and for $i \in \{1, \dots, n-1\}$ that $\overline{\Gamma} \vdash N_i : P^i_{s_i t_i}$ (where $s_i = \alpha_i \left[\vec{\alpha} := \vec{t}\right]$ and $t_i = \beta_i \left[\vec{\alpha} := \vec{t}\right]$ for $i \in \{1, \dots, n\}$).

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\Gamma} \vdash x : \forall\vec{\alpha}(P^1_{\alpha_1\beta_1} \to \dots \to \forall\beta(P^n_{\alpha_n\beta_n} \to \mathbf{false}) \to \mathbf{false})}{\overline{\Gamma} \vdash x\vec{t} : P^1_{s_1t_1} \to \dots \to \forall\beta(P^n_{s_nt_n} \to \mathbf{false}) \to \mathbf{false}} \qquad \overline{\Gamma} \vdash N_1 : P^1_{s_1t_1}}{\vdots}}{\overline{\Gamma} \vdash x\vec{t}\vec{N} : \forall\beta(P^n_{s_nt_n} \to \mathbf{false}) \to \mathbf{false}} \qquad \overline{\Gamma} \vdash L : \forall\beta(P^n_{s_nt_n} \to \mathbf{false})}{\overline{\Gamma} \vdash (x\vec{t}\vec{N})L : \mathbf{false}}$$

For $i \in \{1, \dots, n-1\}$ we can apply Lemma 37 to $\overline{\Gamma} \vdash N_i : P^i_{s_it_i}$ to get that $s_i, t_i \in \mathcal{V}_P$ and that $\Gamma \vdash P^i(s_i, t_i)$. But to proof that $\mathbf{false}$ is a semantic consequence of Γ we still need $\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})$.

To deduce this we have to take a closer look at L. First note that because either $\alpha_n = s_n$ or there exits an $i \in \{1, \dots, n-1\}$ such that $\alpha_n \in \mathrm{FV}(P^i(\alpha_i, \beta_i))$ which implies that $s_n = s_i$ or $s_n = t_i$. In all cases we get that $s_n \in \mathcal{V}_P$. A similar argument yields $t_n \in \mathcal{V}_P$.

If, for some $y \in \mathcal{V}_V$, $l \in \mathbb{N}$, and $s, t, t', t_1, \dots, t_l \in \mathrm{T}_{\lambda 2}$ with $M' := yt_1 \dots t_l$, the term L is equal to y, to $\Lambda\beta.y$, to $\Lambda\beta.M'\,t'$, or to $M'\,t'$ then $y = x_A$ for some universal formula $A = \forall\vec{\alpha}(P^n(s,t) \to \mathbf{false}) \in \Gamma$. It is easy to see that in all three cases we can indeed deduce $\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})$.

If $L = \Lambda\beta.M_1M_2$ for some $M_1, M_2 \in \Lambda_{\mathrm{T}_{\lambda 2}}$ it follows that $M_1 = x_A\vec{t'}N'_1 \dots N'_{n'}$ for some universal formula $A \in \Gamma$, $l \in \mathbb{N}$, some $\vec{t'} = t'_1 \dots t'_l$ where $t'_1, \dots, t'_l \in \mathrm{T}_{\lambda 2}$, $n' \in \mathbb{N}$, and some $N'_1, \dots, N'_{n'} \in \Lambda_{\mathrm{T}_{\lambda 2}}$. We get that $L = \Lambda\beta.x_A\vec{t'}\vec{N'}$ where $\vec{N'} := N'_1 \dots N'_{n'}M_2$. Hence, $\beta \notin \mathrm{FV}(\overline{\Gamma})$, we can use the $\forall$-Introduction rule to deduce $\overline{\Gamma} \vdash x_A\vec{t'}\vec{N'} : P^n_{s_nt_n} \to \mathbf{false}$. Now we can conclude $\Gamma \vdash P^n(s_n, t_n) \to \mathbf{false}$

as in the proof of Lemma 37. Since β is also not in $\mathrm{FV}(\Gamma)$ we can use the $\forall$-Introduction of System **P** to deduce $\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})$ as desired.

If $L = \Lambda\beta.\lambda y : t'.N$ for some $y \in \mathcal{V}_V$, some $t' \in \mathrm{T}_{\lambda 2}$, and some $N \in \Lambda_{\mathrm{T}_{\lambda 2}}$ then $t' = P^n_{s_n t_n}$. Furthermore:

$$\dfrac{\dfrac{\overline{\Gamma}, y : P^n_{s_n t_n} \vdash N : \mathbf{false}}{\overline{\Gamma} \vdash \lambda y : P^n_{s_n t_n}.N : P^n_{s_n t_n} \to \mathbf{false}}}{\overline{\Gamma} \vdash \Lambda\beta.\lambda y : P^n_{s_n t_n}.N : \forall\beta(P^n_{s_n t_n} \to \mathbf{false})}$$

Because $s_n, t_n \in \mathcal{V}_P$ we know that $P^n(s_n, t_n)$ is a valid **P**-formula. So we can apply the induction hypothesis to $\overline{\Gamma}, y : P^n_{s_n t_n} \vdash N : \mathbf{false}$ and it follows that $\Gamma, P^n(s_n, t_n) \vdash \mathbf{false}$. Now we can deduce $\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})$.

$$\dfrac{\dfrac{\Gamma, P^n(s_n, t_n) \vdash \mathbf{false}}{\Gamma \vdash P^n(s_n, t_n) \to \mathbf{false}}}{\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})}$$

All other forms for L (i.e. $M_1 M_2$, $\lambda y : t'.M'$, $\Lambda\beta.\Lambda\gamma.M'$, and $M'\,t'$ with $M' \neq y t_1 \ldots t_l$) are impossible.

Now we can show that **false** is a semantic consequence of Γ.

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false}) \to \mathbf{false}} \qquad \Gamma \vdash \forall\beta(P^n(s_n, t_n) \to \mathbf{false})}{\Gamma \vdash \mathbf{false}}$$

where the elided derivation begins

$$\dfrac{\dfrac{\Gamma \vdash \forall\vec{\alpha}(P^1(\alpha_1, \beta_1) \to \cdots \to \forall\beta(P^n(\alpha_n, \beta_n) \to \mathbf{false}) \to \mathbf{false})}{\Gamma \vdash P^1(s_1, t_1) \to \cdots \to \forall\beta(P^n(s_n, t_n) \to \mathbf{false}) \to \mathbf{false}} \qquad \Gamma \vdash P^1(s_1, t_1)}{\vdots}$$

<u>$M = \lambda x : t_1.M'$</u> for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$, some $x \in \mathcal{V}_V \setminus \mathrm{dom}(\Gamma)$, and some $t_1 \in \mathrm{T}_{\lambda 2}$.
It follows that $\mathbf{false} = t_1 \to t_2$ for some $t_2 \in \mathrm{T}_{\lambda 2}$ which is impossible.

<u>$M = \Lambda\gamma.M'$</u> for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and some $\gamma \in \mathcal{V}_T$.
It follows that $\mathbf{false} = \forall\gamma.t'$ for some $t' \in \mathrm{T}_{\lambda 2}$. Again is a contradiction and makes this case impossible.

<u>$M = M'\,t'$</u> for some $M' \in \Lambda_{\mathrm{T}_{\lambda 2}}$ and some $t' \in \mathrm{T}_{\lambda 2}$.
Since M is in normal form we have that $M' = x M_1 \ldots M_n$ for some $x \in \mathcal{V}_V$, $n \in \mathbb{N}$, and some $M_1, \ldots, M_n \in \Lambda_{\mathrm{T}_{\lambda 2}} \cup \mathrm{T}_{\lambda 2}$. Hence, $(x : \overline{A}) \in \overline{\Gamma}$ for some **P**-formula A and $M = M'\,t'$, we get that this case is impossible because no such A exists.

$\square$

Lemma 39.

$$\Gamma \vdash \boldsymbol{false} \qquad \textit{iff} \qquad \textit{There is a } \boldsymbol{\lambda 2} \textit{ term } M \textit{ such that } \overline{\Gamma} \vdash M : \boldsymbol{false}$$

Proof. The $\Leftarrow$ direction follows from Lemma 38. And the $\Rightarrow$ direction follows from Lemma 36 with $A = \mathbf{false}$. $\square$

Theorem 40. *The inhabitation problem for* $\boldsymbol{\lambda 2}$ *is undecidable.*

Proof. From Lemma 39 it follows that **CONS** $\leq$ **INHAB**. Since , by Theorem 34, **CONS** is undecidable we have shown that **INHAB** is undecidable too. $\square$

References

[1] H.P. Barendregt, Lambda calculi with types, Handbook of Logic in Computer Science, Volume II, 1993.

[2] P. Urzyczyn, Inhabitation in Typed Lambda-Calculi (A Syntactic Approach), Typed Lambda Calculi and Applications, 1997.