

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>2</b>  |
| 1.1      | Conventions . . . . .   | 2         |
| <b>2</b> | <b>Basic Definitions</b>  | <b>2</b>  |
| 2.1      | $\lambda$ -calculus <b><math>\lambda\mathbf{2}</math></b> . . . . . | 2         |
| 2.2      | first-order logic . . . . .   | 6         |
| 2.3      | two-counter automaton . . . . .                                     | 8         |
| <b>3</b> | <b>System P</b>   | <b>9</b>  |
| 3.1      | Definitions . . . . .   | 9         |
| 3.2      | <b>CONS</b> is undecidable . . . . .                                | 11        |
| <b>4</b> | <b>INHAB is undecidable</b>   | <b>19</b> |

# 1 Introduction

## 1.1 Conventions

$\lambda 2$ -types:  $t, t', t'', t_1, t_2, \dots, s, s_1, s_2, \dots$

$\lambda 2$ -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:  $\varphi, \psi,$

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:  $P, P^1, P^2, \dots$

**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:  $Q, Q', \bar{Q}, Q_f, Q_0, Q_1, Q_2, \dots$

Usually we will use Greek letters for bound type-variables and Latin letters for free type-variables.

## 2 Basic Definitions

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

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  $\lambda 2$  types over  $\mathcal{V}_T$ , denoted by  $T_{\lambda 2}$ , is the smallest set  $T$  satisfying the following conditions:

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

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

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

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

**Definition 2.** Let  $M, N \in \Lambda_{T_{\lambda_2}}$  and  $x \in \mathcal{V}_V$ . The substitution of  $x$  by  $N$  in  $M$ , denoted by  $M[x := N]$  is defined as follows:

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

Let  $t, t' \in T_{\lambda_2}$  and  $a \in \mathcal{V}_T$ . The substitution of  $a$  by  $t'$  in  $t$ , denoted by  $t[a := t']$  is defined as follows:

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

Let  $M \in \Lambda_{T_{\lambda_2}}$ ,  $a \in \mathcal{V}_T$ , and  $t \in T_{\lambda_2}$ . The substitution of  $a$  by  $t$  in  $M$ , denoted by  $M[a := t]$  is defined as follows:

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

We will often abbreviate  $(\dots(M[a_n := b_n])\dots)[a_1 := b_1]$  to  $M[\vec{a} := \vec{b}]$  where  $\vec{a} = a_1 \dots a_n$  and  $\vec{b} = b_1 \dots b_n$ .

**Definition 3.** Let  $M \in \Lambda_{T_{\lambda_2}}$ . The set of free variables of  $M$ , denoted by  $\text{FV}(M)$ , is defined inductively as follows:

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

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

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

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

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

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

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

Now we can lift this definition to terms.

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

**Definition 6.** The  $\alpha$ -conversion, denoted by  $\rightarrow_\alpha$ , is a binary relation on  $\Lambda_{T_{\lambda 2}}$ . For all  $M \in \Lambda_{T_{\lambda 2}}$ ,  $x, x' \in \mathcal{V}_T$ ,  $t \in T_{\lambda 2}$ , and  $\alpha, \beta \in \mathcal{V}_T$  if  $x' \notin FV(M) \cup BV(M)$  then  $\lambda x : t.M \rightarrow_\alpha \lambda x' : t.(M[x := x'])$  and if  $\beta \notin FTV(M) \cup BTV(M)$  then  $\Lambda \alpha.M \rightarrow_\alpha \Lambda \beta : t.(M[\alpha := \beta])$ .

The  $\beta$ -reduction, denoted by  $\rightarrow_\beta$ , is a binary relation on  $\Lambda_{T_{\lambda 2}}$ . For all  $M, N \in \Lambda_{T_{\lambda 2}}$ ,  $x \in \mathcal{V}_T$ ,  $t \in T_{\lambda 2}$ , and  $\alpha \in \mathcal{V}_T$  if  $BV(M) \cap FV(N) = \emptyset$  then  $(\lambda x : t.M)N \rightarrow_\beta M[x := N]$  and if  $BTV(M) \cap FTV(N) = \emptyset$  then  $(\Lambda \alpha.M)t \rightarrow_\beta M[\alpha := t]$ .

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

**Definition 7.** So, for  $\rightarrow_\alpha$  we define a new relation  $\Rightarrow_\alpha$ . Let  $M, N, M' \in \Lambda_{T_{\lambda 2}}$ ,  $x \in \mathcal{V}_T$ ,  $t \in T_{\lambda 2}$ , and  $\alpha \in \mathcal{V}_T$ . If  $M \rightarrow_\alpha M'$  then

$$\begin{array}{lll} M \Rightarrow_\alpha M' & MN \Rightarrow_\alpha M'N & \lambda x : t.M \Rightarrow_\alpha \lambda x : t.M' \\ Mt \Rightarrow_\alpha M't & NM \Rightarrow_\alpha NM' & \Lambda \alpha.M \Rightarrow_\alpha \Lambda \alpha.M' \end{array}$$

For  $\rightarrow_\beta$  we define  $\Rightarrow_\beta$  analogously.

**Definition 8.** Let  $M \in \Lambda_{T_{\lambda 2}}$ . The term  $M$  is in normal form if there is no  $N', N \in \Lambda_{T_{\lambda 2}}$  such that  $M \Rightarrow_{\alpha}^* N' \Rightarrow_{\beta} N$ .

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

The term  $M$  is strongly normalizing if there is no infinite chain  $M \Rightarrow_{\alpha}^* M'_1 \Rightarrow_{\beta} M_1 \Rightarrow_{\alpha}^* M'_2 \Rightarrow_{\beta} M_2 \dots$ .

**Definition 9.** Let  $\mathcal{V} = \{x_1, \dots, x_n\}$  be a finite subset of  $\mathcal{V}_V$  and  $t_1, \dots, t_n \in \Lambda_{T_{\lambda 2}}$ . A  $\lambda 2$ -basis  $\Gamma = \{(x_1 : t_1), \dots, (x_n : t_n)\}$  is a mapping from  $\mathcal{V}$  to  $T_{\lambda 2}$ . If the kind of basis is clear from the context we abbreviate  $\lambda 2$ -basis to basis.

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

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

**Definition 10.** Let  $M$  be in  $\Lambda_{T_{\lambda 2}}$ ,  $t$  in  $T_{\lambda 2}$ , and  $\Gamma$  be a basis. A statement  $M : t$  is derivable from  $\Gamma$ , denoted by  $\Gamma \vdash M : t$ , if  $M : t$  can be produced using the following rules.

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

**Definition 11.** A term  $M \in \Lambda_{T_{\lambda 2}}$  is well typed if there exists a basis  $\Gamma$  and a type  $t \in T_{\lambda 2}$  such that  $\Gamma \vdash M : t$  holds.

**Theorem 12.** 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  $\Gamma$  be a basis. If  $\Gamma \vdash M : t$  then  $\Gamma \vdash M' : t$ .

**Theorem 13.** All well typed  $\lambda 2$  terms are strongly normalizing.

**Definition 14.** The inhabitation problem for  $\lambda 2$ , denoted by **INHAB**, is defined as follows. Given a  $\lambda 2$  type  $t$ .

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

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

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

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

## 2.2 first-order logic

**Definition 15.** A ranked set is a tuple  $(\Sigma, rk)$ , where  $\Sigma$  is a countable set and  $rk: \Sigma \rightarrow \mathbb{N}$  is a function that maps every symbol from  $\Sigma$  to a natural number (its rank).

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

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

**Definition 16.** 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  $t_1, t_2, \dots, t_k \in \mathcal{T}$  then  $f(t_1, t_2, \dots, t_k) \in \mathcal{T}$ .

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

We introduce an additional binary operation  $\rightarrow$  on formulas, where for some  $\varphi, \psi \in \mathcal{L}_{(\mathcal{V}, \mathcal{F}, \mathcal{P})}$  the formula  $(\varphi \rightarrow \psi)$  is defined as  $(\neg \varphi \vee \psi)$ , if we have a formula of the form  $(\varphi_1 \rightarrow (\varphi_2 \rightarrow (\dots \rightarrow (\varphi_{n-1} \rightarrow \varphi_n) \dots)))$  we will often omit the brackets and just write  $(\varphi_1 \rightarrow \varphi_2 \rightarrow \dots \rightarrow \varphi_{n-1} \rightarrow \varphi_n)$  or  $\varphi_1 \rightarrow \varphi_2 \rightarrow \dots \rightarrow \varphi_{n-1} \rightarrow \varphi_n$  instead.

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

**Definition 17.** The set of variables of a term  $t \in \mathcal{T}_{(\mathcal{V}, \mathcal{F})}$ , denoted by  $V(t)$ , is defined by:

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

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

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

**Definition 18.** The set of subformulas of a formula  $\varphi \in \mathcal{L}_{(\mathcal{V}, \mathcal{F}, \mathcal{P})}$ , denoted by  $\text{SUB}(\varphi)$ , is defined as follows:

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

**Definition 19.** We say that a formula  $\varphi \in \mathcal{L}_{(\mathcal{V}, \mathcal{F}, \mathcal{P})}$  contains no dummy quantifiers if for all  $\psi \in \text{SUB}(\varphi)$  of the form  $\psi = \forall y.\psi'$  or  $\psi = \exists y.\psi'$  for some  $y \in \mathcal{V}_P$  and some  $\psi' \in \mathcal{L}_{(\mathcal{V}, \mathcal{F}, \mathcal{P})}$  we have that  $y \in \text{FV}(\psi')$ .

**Definition 20.** 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:

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

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

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

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

**Definition 21.** An interpretation  $I$  over  $\mathcal{V}$ ,  $\mathcal{F}$ , and  $\mathcal{P}$  is a triple  $I = (\Delta, \cdot^I, \omega)$ , where

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

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

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

**Definition 22.** 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:

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

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

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

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

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

**Definition 23.** Let  $\Gamma$  be a finite set of first-order formulas.

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

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

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

### 2.3 two-counter automaton

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



**Definition 24.** 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  $\Rightarrow_M^r$  is a binary relation on the configurations of  $M$  such that two configurations  $\langle Q, m, n \rangle, \langle \hat{Q}, \hat{m}, \hat{n} \rangle$  of  $M$  are 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  $\hat{Q} = Q', \hat{m} = m + 1$ , and  $\hat{n} = n$ ,
- if  $r = +(2, Q')$  for some  $Q' \in \mathcal{Q}$  then  $\hat{Q} = Q', \hat{m} = m$ , and  $\hat{n} = n + 1$ ,
- if  $r = -(1, Q_1, Q_2)$  for some  $Q_1, Q_2 \in \mathcal{Q}$  then
  - if  $m = 0$  then  $\hat{Q} = Q_2, \hat{m} = 0$ , and  $\hat{n} = n$ ,
  - if  $m \geq 1$  then  $\hat{Q} = Q_1, \hat{m} = m - 1$ , and  $\hat{n} = n$ ,
- if  $r = -(2, Q_1, Q_2)$  for some  $Q_1, Q_2 \in \mathcal{Q}$  then
  - if  $n = 0$  then  $\hat{Q} = Q_2, \hat{m} = m$ , and  $\hat{n} = 0$ ,
  - if  $n \geq 1$  then  $\hat{Q} = Q_1, \hat{m} = m$ , and  $\hat{n} = n - 1$ .

The transition relation of  $M$ , denoted by  $\Rightarrow_M$ , is defined as  $\bigcup_{r \in R(\mathcal{Q} \setminus \{Q_f\})} \Rightarrow_M^r$ . 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  $\hat{m}, \hat{n} \in \mathbb{N}$  such that  $\langle Q_0, m, n \rangle \Rightarrow_M^* \langle Q_f, \hat{m}, \hat{n} \rangle$  (It follows that there exists an  $i \in \mathbb{N}$  and configurations  $D_1, \dots, D_i$  of  $M$  such that  $\langle Q_0, m, n \rangle = D_1 \Rightarrow_M \dots \Rightarrow_M D_i = \langle Q_f, \hat{m}, \hat{n} \rangle$ , we call this chain a computation with length  $i$ ).

**Definition 25.** The halting problem for two-counter automaton, denoted by **HALT**, is defined as follows. Given a two-counter automaton  $M$ .

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

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

## 3 System P

### 3.1 Definitions

In the following let  $\mathcal{V}_P = \{\alpha, a, \beta, b, \dots\}$  be a countably infinite subset of  $\mathcal{V}_T$  (of variables). Let  $\mathcal{P}_P = \{P, Q, \dots\}$  be a set (of predicate symbols) and  $\mathcal{P}$  a ranked set such

that  $\mathcal{P}^{(0)} = \{\mathbf{false}\}$ ,  $\mathcal{P}^{(2)} = \mathcal{P}_P$ , and  $\mathcal{P}^{(k)} = \emptyset$  for all  $k \in \mathbb{N} \setminus \{0, 2\}$ . A first-order logic formula  $\varphi$  over  $\mathcal{V}_P, \emptyset$ , 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  $\varphi = \forall \vec{\alpha}(A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n)$  for some  $n \in \mathbb{N}$  and where  $A_i$  is an atomic formula for  $i \in \{1, \dots, n\}$ ,  $A_i \neq \mathbf{false}$  for  $i \in \{1, \dots, n-1\}$  and for each  $\alpha \in \text{FV}(A_n) \setminus \text{FV}(\varphi)$  there exists an  $i \in \{1, \dots, n-1\}$  such that  $\alpha \in \text{FV}(A_i)$ .

**existential formula** if there is 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  $\alpha \in (\text{FV}(A_n) \setminus \text{FV}(\varphi)) \setminus \{\beta\}$  there exists an  $i \in \{1, \dots, n-1\}$  such that  $\alpha \in \text{FV}(A_i)$  and  $\varphi = \forall \vec{\alpha}(A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_{n-1} \rightarrow \forall \beta(A_n \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$ .

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

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

**Definition 27.** Let  $A$  be a **P**-formula, and  $\Gamma$  be a basis. The formula  $A$  is a semantic consequence of  $\Gamma$ , denoted by  $\Gamma \vdash A$ , if  $A$  can be produced using the following deduction rules.

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

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 28.** Let  $\Gamma$  be a basis. The **P**-formula  $A$  is a semantic consequence with falsity of  $\Gamma$ , denoted by  $\Gamma \vdash_f A$ , 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.

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

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

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

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  $B$ .  $\square$

**Definition 29.** The problem to decide whether a given set of **P**-formulas is consistent, denoted by **CONS**, is defined as follows. Given a set of **P**-formulas  $\Gamma$ .

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

### 3.2 CONS is undecidable

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

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

Let  $M = (\mathcal{Q}, Q_0, Q_f, R)$  be a two-counter automaton, w.l.o.g.  $S, P, R_1, R_2, E, D, G \notin \mathcal{Q}$ . 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 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  $\Gamma_C$ . It contains the following formulas:

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

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

- $\forall \alpha \beta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow Q'(\beta))$   
change of state
- $\forall \alpha \beta \gamma \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow R_1(\beta, \delta) \rightarrow P(\delta, \gamma))$   
increment register 1
- $\forall \alpha \beta \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\beta, \delta) \rightarrow D(\delta))$   
prevent zero in register 1
- $\forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_2(\alpha, \gamma) \rightarrow R_2(\beta, \gamma))$   
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:

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

- $\forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow E(\gamma) \rightarrow R_1(\beta, \gamma))$   
if register 1 is zero it stays zero
- $\forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow D(\gamma) \rightarrow Q_1(\beta))$   
change state to  $Q_1$  if register 1 is greater zero
- $\forall \alpha \beta \gamma \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow D(\gamma) \rightarrow P(\gamma, \delta) \rightarrow R_1(\beta, \delta))$   
decrement register 1 if possible
- $\forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_2(\alpha, \gamma) \rightarrow R_2(\beta, \gamma))$   
do not change register 2 in both cases

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

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

- $\forall \alpha \beta (S(\alpha, \beta) \rightarrow G(\beta))$
- $\forall \alpha (D(\alpha) \rightarrow G(\alpha))$
- $\forall \alpha (G(\alpha) \rightarrow \forall \beta (R_1(\alpha, \beta) \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$   
every element represents a configuration so it has a value for register 1
- $\forall \alpha (G(\alpha) \rightarrow \forall \beta (R_2(\alpha, \beta) \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$   
every element represents a configuration so it has a value for register 2
- $\forall \alpha (G(\alpha) \rightarrow \forall \beta (S(\alpha, \beta) \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$   
every element has a successor

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

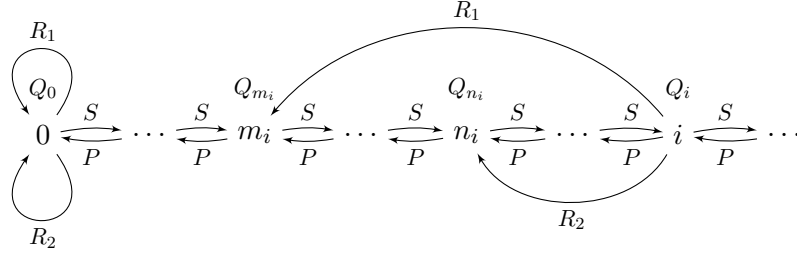
**Claim 30.**

$$\Gamma_M \vdash \mathbf{false} \text{ holds in system } P \quad \implies \quad M \text{ terminates on input } (0, 0)$$

*Proof.* Assume  $M$  does not terminate then there is an infinite chain  $C_0 \Rightarrow_M C_1 \Rightarrow_M C_2 \Rightarrow_M \dots$  ( $C_i = \langle Q_i, m_i, n_i \rangle$  for  $i \in \mathbb{N}$ ). Now we construct a model of  $\Gamma_M$  which interprets **false** with  $\perp$  this contradicts  $\Gamma_M \vdash \mathbf{false}$ .

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  $i^{\text{th}}$  configuration of our infinite computation. Now the idea for our model of  $\Gamma_M$  looks like this:



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

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

$$a^I = 0$$

$$a_0^I = 0$$

$$b_0^I = 0$$

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

We proof the other direction by induction on the length of the computation. But to be able to use the induction hypothesis we need a slightly more general statement (this is why we defined  $\Gamma_{\overline{M}}$  and not just  $\Gamma_M$  right away).

**Claim 31.** Let  $C = \langle Q, m, n \rangle$  be a configuration of  $M$ . If a final configuration (i.e. a configuration  $\langle Q_f, \hat{m}, \hat{n} \rangle$  for some  $\hat{m}, \hat{n} \in \mathbb{N}$ ) is reachable from  $C$  then  $\Gamma_C \cup \Gamma_{\overline{M}} \vdash \mathbf{false}$ .

*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  $\Gamma_C$  for some  $a \in \mathcal{V}_P$  and  $\forall \alpha (Q_f(\alpha) \rightarrow \mathbf{false})$  is in  $\Gamma_{\overline{M}}$ , we can easily deduce  $\mathbf{false}$ .

$$\frac{\frac{\Gamma_C \cup \Gamma_{\overline{M}} \vdash \forall \alpha (Q_f(\alpha) \rightarrow \mathbf{false})}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash Q_f(a) \rightarrow \mathbf{false}} \quad \Gamma_C \cup \Gamma_{\overline{M}} \vdash Q_f(a)}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash \mathbf{false}}$$

Induction Step:  $i = i' + 1$

Since  $I \models \mathbf{false}$  holds trivially if  $I$  interprets  $\mathbf{false}$  with  $\top$  we only need to consider

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

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

$$\begin{aligned} &R_1(a, a_0), P(a_{i-1}, a_i), G(a_{i-1}), \text{ and } D(a_{i-1}) \text{ for } i \in \{1, \dots, m\}, \\ &R_2(a, b_0), P(b_{i-1}, b_i), G(b_{i-1}), \text{ and } D(b_{i-1}) \text{ for } i \in \{1, \dots, n\}, \\ &Q(a), E(a_m), E(b_n), G(a), G(a_m), \text{ and } G(b_n). \end{aligned}$$

And  $\Gamma_D$  contains the formulas:

$$\begin{aligned} &R_1(\widehat{a}, \widehat{a}_0), P(\widehat{a}_{i-1}, \widehat{a}_i), G(\widehat{a}_{i-1}), \text{ and } D(\widehat{a}_{i-1}) \text{ for } i \in \{1, \dots, \widehat{m}\}, \\ &R_2(\widehat{a}, \widehat{b}_0), P(\widehat{b}_{i-1}, \widehat{b}_i), G(\widehat{b}_{i-1}), \text{ and } D(\widehat{b}_{i-1}) \text{ for } i \in \{1, \dots, \widehat{n}\}, \\ &\widehat{Q}(\widehat{a}), E(\widehat{a}_{\widehat{m}}), E(\widehat{b}_{\widehat{n}}), G(\widehat{a}), G(\widehat{a}_{\widehat{m}}), \text{ and } G(\widehat{b}_{\widehat{n}}). \end{aligned}$$

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

$$\frac{\text{Induction Hypothesis} \quad \frac{\Gamma_C \cup \Gamma_{\overline{M}} \cup \Gamma_D \vdash_f \mathbf{false} \quad \Gamma_C \cup \Gamma_{\overline{M}} \vdash_f \Gamma_D}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash_f \mathbf{false}}}{\Gamma_C \cup \Gamma_{\overline{M}} \vdash_f \mathbf{false}}$$

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.

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

We first introduce a new variable representing the new configuration  $D$  (let  $b \in \mathcal{V}_P \setminus \text{FV}(\Gamma_C)$ , note that  $\text{FV}(\Gamma_{\overline{M}}) = \emptyset$ ).

$$\frac{\begin{array}{c} \vdots \\ \hline S(a, b) \vdash_f \mathbf{f} \end{array} \quad \frac{\vdash_f \forall \alpha (G(\alpha) \rightarrow \forall \beta (S(a, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f})}{\vdash_f G(a) \rightarrow (\forall \beta (S(a, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f})} \quad \vdash_f G(a)}{\frac{\vdash_f (\forall \beta (S(a, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f}) \rightarrow \mathbf{f} \quad \vdash_f \forall \beta (S(a, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f}}{\vdash_f \mathbf{f}}}$$

For the new variable  $b$  we have to deduce  $G(b)$ .

$$\frac{\frac{\vdots}{G(b) \vdash_f \mathbf{f}} \quad \frac{\frac{\vdash_f \forall \alpha \beta (S(\alpha, \beta) \rightarrow G(\beta))}{\vdash_f S(a, b) \rightarrow G(b)} \quad \vdash_f S(a, b)}{\vdash_f G(b) \rightarrow \mathbf{f}} \quad \vdash_f G(b)}{\vdash_f \mathbf{f}}$$

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

$$\frac{\frac{\vdots}{R_2(b, b_0) \vdash_f \mathbf{f}} \quad \frac{\frac{\vdash_f \forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_2(\alpha, \gamma) \rightarrow R_2(\beta, \gamma))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_2(a, b_0) \rightarrow R_2(b, b_0)} \quad \vdash_f Q(a)}{\vdash_f S(a, b) \rightarrow R_2(a, b_0) \rightarrow R_2(b, b_0)} \quad \vdash_f S(a, b)}{\vdash_f R_2(a, b_0) \rightarrow R_2(b, b_0)} \quad \vdash_f R_2(a, b_0)}{\vdash_f R_2(b, b_0)} \quad \vdash_f R_2(b, b_0)}{\vdash_f \mathbf{f}}$$

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

$$\frac{\frac{\vdots}{Q'(b) \vdash_f \mathbf{f}} \quad \frac{\frac{\vdash_f \forall \alpha \beta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow Q'(\beta))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow Q'(b)} \quad \vdash_f Q(a)}{\vdash_f S(a, b) \rightarrow Q'(b)} \quad \vdash_f S(a, b)}{\vdash_f Q'(b)} \quad \vdash_f Q'(b)}{\vdash_f \mathbf{f}}$$

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

$$\frac{\frac{\vdots}{R_1(b, d) \vdash_f \mathbf{f}} \quad \frac{\vdash_f \forall \beta (R_1(b, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f} \vdash_f \mathbf{f}}{\vdash_f (\forall \beta (R_1(b, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f}) \rightarrow \mathbf{f}} \quad \frac{\vdash_f \forall \alpha (\forall \beta (R_1(\alpha, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f})}{\vdash_f \forall \beta (R_1(b, \beta) \rightarrow \mathbf{f}) \rightarrow \mathbf{f}}}{\vdash_f \mathbf{f}}$$

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



$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \gamma \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow R_1(\beta, \delta) \rightarrow P(\delta, \gamma))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(a, a_0) \rightarrow R_1(b, d) \rightarrow P(d, a_0)} \quad \vdash_f Q(a) \\
\frac{\vdash_f S(a, b) \rightarrow R_1(a, a_0) \rightarrow R_1(b, d) \rightarrow P(d, a_0)}{\vdash_f R_1(a, a_0) \rightarrow R_1(b, d) \rightarrow P(d, a_0)} \quad \vdash_f S(a, b) \\
\vdots \\
\frac{P(d, a_0) \vdash_f \mathbf{f}}{\vdash_f P(d, a_0) \rightarrow \mathbf{f}} \quad \frac{\vdash_f R_1(b, d) \rightarrow P(d, a_0) \quad \vdash_f R_1(b, d)}{\vdash_f P(d, a_0)} \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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

$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\beta, \delta) \rightarrow D(\delta))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(b, d) \rightarrow D(d)} \quad \vdash_f Q(a) \\
\frac{\vdash_f S(a, b) \rightarrow R_1(b, d) \rightarrow D(d)}{\vdash_f R_1(b, d) \rightarrow D(d)} \quad \vdash_f S(a, b) \\
\vdots \\
\frac{D(d) \vdash_f \mathbf{f}}{\vdash_f D(d) \rightarrow \mathbf{f}} \quad \frac{\vdash_f R_1(b, d) \rightarrow D(d) \quad \vdash_f R_1(b, d)}{\vdash_f D(d)} \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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

$$\begin{array}{c}
\vdots \\
\frac{G(d) \vdash_f \mathbf{f}}{\vdash_f G(d) \rightarrow \mathbf{f}} \quad \frac{\vdash_f \forall \alpha (D(\alpha) \rightarrow G(\alpha))}{\vdash_f D(d) \rightarrow G(d)} \quad \vdash_f D(d) \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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

The other case, that  $\mathbf{r} = -(\mathbf{Q}, \mathbf{1}, \mathbf{Q}_1, \mathbf{Q}_2)$ , has to be split into two cases again. If  $\mathbf{m} = \mathbf{0}$  then  $\hat{Q} = Q_2$ ,  $\hat{m} = 0$ , and  $\hat{n} = n$ . We only need to ensure that the successor state is  $Q_2$  and that register 1 is still zero.

$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow E(\gamma) \rightarrow Q_2(\beta))}{\frac{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(a, a_0) \rightarrow E(a_0) \rightarrow Q_2(b) \quad \vdash_f Q(a)}{\vdash_f S(a, b) \rightarrow R_1(a, a_0) \rightarrow E(a_0) \rightarrow Q_2(b)} \quad \vdash_f S(a, b)} \\
\vdots \\
\frac{Q_2(b) \vdash_f \mathbf{f}}{\vdash_f Q_2(b) \rightarrow \mathbf{f}} \quad \frac{\vdash_f R_1(a, a_0) \rightarrow E(a_0) \rightarrow Q_2(b) \quad \vdash_f R_1(a, a_0)}{\vdash_f E(a_0) \rightarrow Q_2(b)} \quad \vdash_f E(a_0) \\
\hline
\vdash_f Q_2(b)
\end{array}$$

Register 1 stays zero.

$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow E(\gamma) \rightarrow R_1(\beta, \gamma))}{\frac{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(a, a_0) \rightarrow E(a_0) \rightarrow R_1(b, a_0) \quad \vdash_f Q(a)}{\vdash_f S(a, b) \rightarrow R_1(a, a_0) \rightarrow E(a_0) \rightarrow R_1(b, a_0) \quad \vdash_f S(a, b)}} \\
\vdots \\
\frac{R_1(b, a_0) \vdash_f \mathbf{f}}{\vdash_f R_1(b, a_0) \rightarrow \mathbf{f}} \quad \frac{\vdash_f R_1(a, a_0) \rightarrow E(a_0) \rightarrow R_1(b, a_0) \quad \vdash_f R_1(a, a_0)}{\vdash_f E(a_0) \rightarrow R_1(b, a_0) \quad \vdash_f E(a_0)} \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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  $\Gamma_D$ . So we can use the induction hypothesis to deduce **false**.

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

$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \gamma (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow D(\gamma) \rightarrow Q_1(\beta))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(a, a_0) \rightarrow D(a_0) \rightarrow Q_1(b)} \quad \vdash_f Q(a) \\
\frac{\vdash_f S(a, b) \rightarrow R_1(a, a_0) \rightarrow D(a_0) \rightarrow Q_1(b)}{\vdash_f R_1(a, a_0) \rightarrow D(a_0) \rightarrow Q_1(b)} \quad \vdash_f S(a, b) \\
\vdots \\
\frac{Q_1(b) \vdash_f \mathbf{f}}{\vdash_f Q_1(b) \rightarrow \mathbf{f}} \quad \frac{\vdash_f D(a_0) \rightarrow Q_1(b) \quad \vdash_f D(a_0)}{\vdash_f Q_1(b)} \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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

$$\begin{array}{c}
\frac{\vdash_f \forall \alpha \beta \gamma \delta (Q(\alpha) \rightarrow S(\alpha, \beta) \rightarrow R_1(\alpha, \gamma) \rightarrow D(\gamma) \rightarrow P(\gamma, \delta) \rightarrow R_1(\beta, \delta))}{\vdash_f Q(a) \rightarrow S(a, b) \rightarrow R_1(a, a_0) \rightarrow D(a_0) \rightarrow P(a_0, a_1) \rightarrow R_1(b, a_1)} \quad \vdash_f Q(a) \\
\frac{\vdash_f S(a, b) \rightarrow R_1(a, a_0) \rightarrow D(a_0) \rightarrow P(a_0, a_1) \rightarrow R_1(b, a_1)}{\vdash_f R_1(a, a_0) \rightarrow D(a_0) \rightarrow P(a_0, a_1) \rightarrow R_1(b, a_1)} \quad \vdash_f S(a, b) \\
\vdots \\
\frac{R_1(b, a_1) \vdash_f \mathbf{f}}{\vdash_f R_1(b, a_1) \rightarrow \mathbf{f}} \quad \frac{\vdash_f D(a_0) \rightarrow P(a_0, a_1) \rightarrow R_1(b, a_1) \quad \vdash_f D(a_0)}{\vdash_f P(a_0, a_1) \rightarrow R_1(b, a_1)} \quad \vdash_f P(a_0, a_1) \\
\hline
\vdash_f \mathbf{f}
\end{array}$$

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

**Lemma 32.**

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

*Proof.* The  $\Leftarrow$  direction is proven in Claim 30. And the  $\Rightarrow$  direction is a direct consequence of Claim 31 with  $C = \langle Q_0, 0, 0 \rangle$ .  $\square$

**Theorem 33.** *CONS is undecidable.*

*Proof.* Since by Lemma 32 for a given two-counter automaton  $M$  we can effectively construct a set of **P**-formulas  $\Gamma_M$  such that  $M$  terminates on input  $(0, 0)$  iff  $\Gamma_M$  is not consistent. It follows that **HALT**  $\leq$  **CONS**. Since **HALT** is undecidable we have shown that **CONS** is undecidable too.  $\square$

## 4 INHAB is undecidable

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

$$\Gamma \vdash \mathbf{false} \quad \text{iff} \quad \text{There is a } \lambda 2 \text{ term } M \text{ such that } \bar{\Gamma} \vdash M : \mathbf{false}$$

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

**Definition 34.** For a **P**-formula  $A$  we define the code of  $A$ , denoted by  $\bar{A}$ , as:

If  $A$  is an atomic formula then

$$\bar{A} = \begin{cases} \mathbf{false} & \text{if } A = \mathbf{false} \\ (\alpha \rightarrow p_1) \rightarrow (\beta \rightarrow p_2) \rightarrow p & \text{if } A = P(\alpha, \beta) \end{cases}$$

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

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

$$\bar{A} = \forall \vec{\alpha} (\bar{A}_1 \rightarrow \bar{A}_2 \rightarrow \dots \rightarrow \bar{A}_n)$$

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

$$\bar{A} = \forall \vec{\alpha} (\bar{A}_1 \rightarrow \dots \rightarrow \forall \beta (\bar{A}_n \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$$

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

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

**Lemma 35.** *Let  $\Gamma$  be a **P**-basis and  $A$  a **P**-formula such that  $\Gamma \vdash A$ . Then there exists a term  $M \in \Lambda_{T_{\lambda 2}}$  such that  $\bar{\Gamma} \vdash M : \bar{A}$  holds.*

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

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

$A$  is produced by the  $\rightarrow$ -Introduction rule. It follows that  $A = A' \rightarrow B'$  for some **P**-formulas  $A'$  and  $B'$ . We can now apply the induction hypothesis to  $\Gamma, A' \vdash B'$  and we get that there exists an  $M' \in \Lambda_{T_{\lambda 2}}$  such that  $\bar{\Gamma}, \bar{A}' \vdash M' : \bar{B}'$ . With the  $\lambda$ -Introduction rule we deduce  $\bar{\Gamma} \vdash \lambda x : \bar{A}'. M' : \bar{A}' \rightarrow \bar{B}'$ . Since  $A$  has to be a universal or an existential formula  $\bar{A}' \rightarrow \bar{B}' = \bar{A} \rightarrow \bar{B}$ . So  $M := \lambda x : \bar{A}'. M'$  has the required type.

$A$  is produced by the  $\rightarrow$ -Elimination rule. So there exists a **P**-formula  $B$  such that  $\Gamma \vdash B \rightarrow A$  and  $\Gamma \vdash B$ . Now we apply the induction hypothesis and get that there exist  $M_1, M_2 \in \Lambda_{T_{\lambda 2}}$  such that  $\bar{\Gamma} \vdash M_1 : \bar{B} \rightarrow \bar{A}$  and  $\bar{\Gamma} \vdash M_2 : \bar{B}$ . Again we have that  $\bar{B} \rightarrow \bar{A} = \bar{B} \rightarrow \bar{A}$ . It follows that  $M := M_1 M_2$  has the type  $\bar{A}$ .

$A$  is produced by the  $\forall$ -Introduction rule. It follows that  $A = \forall \beta B$  for some  $\beta \in \mathcal{V}_P \setminus \text{FV}(\Gamma)$  and some **P**-formula  $B$ . By applying the induction hypothesis to  $\Gamma \vdash B$  we get that there exists an  $M' \in \Lambda_{T_{\lambda 2}}$  such that  $\bar{\Gamma} \vdash M' : \bar{B}$ . We deduce that  $M := \Lambda \beta. M'$  has type  $\forall \beta. \bar{B} = \bar{\forall \beta B}$  as desired.

$A$  is produced by the  $\forall$ -Elimination rule. There is a **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 exists an  $M' \in \Lambda_{T_{\lambda 2}}$  such that  $\bar{\Gamma} \vdash M' : \bar{\forall \alpha B}$ . Since  $\bar{\forall \alpha B} = \forall \alpha. \bar{B}$  the term  $M := M' b$  has the type  $\bar{B}[\alpha := b] = \bar{B}[\alpha := b]$ .

□

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

**Lemma 36.** *Let  $\Gamma$  be a  $\mathbf{P}$ -basis,  $M \in \Lambda_{T_{\lambda 2}}$ ,  $P \in \mathcal{P}_P$ , and  $s, t \in T_{\lambda 2}$  such that  $\bar{\Gamma} \vdash M : P_{st}$  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 all well typed  $\lambda 2$  terms are strongly normalizing (see ). Hence,  $M$  is well typed in  $\lambda 2$ , we can assume that  $M$  is in normal form.

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

$\underline{M = x}$  for some  $y \in \mathcal{V}_V$ .

It follows that  $(x : P_{st}) \in \bar{\Gamma}$ . Now the definition of  $\bar{\Gamma}$  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  $M_1, M_2 \in \Lambda_{T_{\lambda 2}}$ .

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  $N_1, \dots, N_k \in \Lambda_{T_{\lambda 2}} \cup T_{\lambda 2}$ .

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

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

$$\frac{\bar{\Gamma} \vdash x : \forall \vec{\alpha} (P_{\alpha_1 \beta_1}^1 \rightarrow \dots \rightarrow P_{\alpha_n \beta_n}^n \rightarrow P_{\alpha \beta})}{\frac{\bar{\Gamma} \vdash x \vec{t} : P_{s_1 t_1}^1 \rightarrow \dots \rightarrow P_{s_n t_n}^n \rightarrow P_{st} \quad \bar{\Gamma} \vdash N_1 : P_{s_1 t_1}^1}{\vdots} \quad \frac{\bar{\Gamma} \vdash x \vec{t} N_1 \dots N_{n-1} : P_{s_n t_n}^n \rightarrow P_{st} \quad \bar{\Gamma} \vdash N_n : P_{s_n t_n}^n}{\bar{\Gamma} \vdash (x \vec{t} N_1 \dots N_{n-1}) N_n : P_{st}}}$$

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

If  $\alpha = \bar{\alpha}_j$  for some  $j \in \{1, \dots, n\}$  then because there are no dummy quantifiers we get that  $s = \bar{t}_j$ . Furthermore since  $\alpha \in \text{FV}(P(\alpha, \beta)) \setminus \text{FV}(A)$  it follows that there exists an  $i \in \{1, \dots, n\}$  such that  $\alpha \in \text{FV}(P^i(\alpha_i, \beta_i))$ , 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  $\alpha \neq \bar{\alpha}_j$  for all  $j \in \{1, \dots, n\}$  then  $\alpha \in \text{FV}(A)$  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  $\Gamma$ .

$$\begin{array}{c}
\frac{\Gamma \vdash \forall \vec{\alpha}(P^1(\alpha_1, \beta_1) \rightarrow \cdots \rightarrow P^n(\alpha_n, \beta_n) \rightarrow P(\alpha, \beta))}{\Gamma \vdash P^1(s_1, t_1) \rightarrow \cdots \rightarrow P^n(s_n, t_n) \rightarrow P(s, t)} \quad \Gamma \vdash P^1(s_1, t_1) \\
\vdots \\
\frac{\Gamma \vdash P^n(s_n, t_n) \rightarrow P(a, b) \quad \Gamma \vdash P^n(s_n, t_n)}{\Gamma \vdash P(s, t)}
\end{array}$$

$\underline{M = \lambda x : t'.M'}$  for some  $M' \in \Lambda_{T_{\lambda_2}}$ , some  $x \in \mathcal{V}_V \setminus \text{dom}(\Gamma)$ , and some  $t' \in T_{\lambda_2}$ . It follows that  $t' = s \rightarrow p_1$  and  $\bar{\Gamma}, x : s \rightarrow p_1 \vdash M' : (t \rightarrow p_2) \rightarrow p$ .

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

If  $M' = \lambda y : t \rightarrow p_2. zxy$  for some  $y, z \in \mathcal{V}_T$  then  $z = x_{P(s,t)}$  and therefore  $(z : (s \rightarrow p_1) \rightarrow (t \rightarrow p_2) \rightarrow p) \in \bar{\Gamma}$ . 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  $\bar{A}$  has the required type.

$\underline{M = \Lambda \gamma. M'}$  for some  $M' \in \Lambda_{T_{\lambda_2}}$  and some  $\gamma \in \mathcal{V}_T$ .

It follows that  $\bar{\Gamma} \vdash M : \forall \gamma. t'$  for some  $t' \in T_{\lambda_2}$ . But this can not be since  $P_{st} = (s \rightarrow p_1) \rightarrow (t \rightarrow p_2) \rightarrow p$ . Therefore  $M$  is not of the form  $\Lambda \gamma. M'$  and this case is impossible.

$\underline{M = M' t'}$  for some  $M' \in \Lambda_{T_{\lambda_2}}$  and some  $t' \in T_{\lambda_2}$ .

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  $M_1, \dots, M_n \in \Lambda_{T_{\lambda_2}} \cup T_{\lambda_2}$ . Hence,  $(x : \bar{A}) \in \bar{\Gamma}$  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 an existential formula and  $M_1, \dots, M_n \in T_{\lambda_2}$ . 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  $\gamma \in \text{FV}(P(\alpha, \beta)) \setminus \text{FV}(A)$ .

□

**Lemma 37.** *Let  $\Gamma$  be a **P**-basis,  $M \in \Lambda_{T_{\lambda_2}}$  such that  $\bar{\Gamma} \vdash M : \mathbf{false}$  holds. Then  $\Gamma \vdash \mathbf{false}$  holds.*

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

$\underline{M = x}$  for some  $x \in \mathcal{V}_V$ .

It follows that  $(x : \mathbf{false}) \in \bar{\Gamma}$ . Now the definition of  $\bar{\Gamma}$  yields that  $\mathbf{false} \in \Gamma$ . Therefore  $\Gamma \vdash \mathbf{false}$  holds.

$M = M_1 M_2$  for some  $M_1, M_2 \in \Lambda_{T_{\lambda_2}}$ .

Because  $M$  is in normal form we have that  $M_1 = x N_1 \dots N_k$  for some  $x \in \mathcal{V}_T$ ,  $k \in \mathbb{N}$ , and some  $N_1, \dots, N_k \in \Lambda_{T_{\lambda_2}} \cup T_{\lambda_2}$ . 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  $A = \forall \vec{\alpha} (P^1(\alpha_1, \beta_1) \rightarrow \dots \rightarrow P^n(\alpha_n, \beta_n) \rightarrow \mathbf{false})$  where  $\vec{\alpha} = \bar{\alpha}_1 \dots \bar{\alpha}_m$  for some  $m \in \mathbb{N}$  and some  $\bar{\alpha}_1, \dots, \bar{\alpha}_m \in \mathcal{V}_P$ . In this case  $M = x \vec{t} \vec{N}$  for some  $\vec{t} = \bar{t}_1 \dots \bar{t}_m$  with  $\bar{t}_1, \dots, \bar{t}_m \in T_{\lambda_2}$  and some  $\vec{N} = N_1 \dots N_n$  with  $N_1, \dots, N_n \in \Lambda_{T_{\lambda_2}}$ . 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  $A = \forall \vec{\alpha} (P^1(\alpha_1, \beta_1) \rightarrow \dots \rightarrow \forall \beta (P^n(\alpha_n, \beta_n) \rightarrow \mathbf{false}) \rightarrow \mathbf{false})$  where  $\vec{\alpha} = \bar{\alpha}_1 \dots \bar{\alpha}_m$  for some  $m \in \mathbb{N}$  and some  $\bar{\alpha}_1, \dots, \bar{\alpha}_m \in \mathcal{V}_P$  (w.l.o.g.  $\beta \notin \text{FV}(\bar{\Gamma})$  and  $\beta \neq \bar{\alpha}_i$  for all  $i \in \{1, \dots, m\}$ ). Then  $M$  has to be of the form  $M = x \vec{t} \vec{N} L$  for some  $\vec{t} = \bar{t}_1 \dots \bar{t}_m$  with  $\bar{t}_1, \dots, \bar{t}_m \in T_{\lambda_2}$ , some  $\vec{N} = N_1 \dots N_{n-1}$  with  $N_1, \dots, N_{n-1} \in \Lambda_{T_{\lambda_2}}$ , and some  $L \in \Lambda_{T_{\lambda_2}}$ . It also has to hold that  $\bar{\Gamma} \vdash L : \forall \beta (P_{s_n t_n}^n \rightarrow \mathbf{false})$  and for  $i \in \{1, \dots, n-1\}$  that  $\bar{\Gamma} \vdash N_i : P_{s_i t_i}^i$  (where  $s_i = \alpha_i [\vec{\alpha} := \vec{t}]$  and  $t_i = \beta_i [\vec{\alpha} := \vec{t}]$  for  $i \in \{1, \dots, n\}$ ).

$$\begin{array}{c}
\bar{\Gamma} \vdash x : \forall \vec{\alpha} (P_{\alpha_1 \beta_1}^1 \rightarrow \dots \rightarrow \forall \beta (P_{\alpha_n \beta_n}^n \rightarrow \mathbf{false}) \rightarrow \mathbf{false}) \\
\hline
\bar{\Gamma} \vdash x \vec{t} : P_{s_1 t_1}^1 \rightarrow \dots \rightarrow \forall \beta (P_{s_n t_n}^n \rightarrow \mathbf{false}) \rightarrow \mathbf{false} \quad \bar{\Gamma} \vdash N_1 : P_{s_1 t_1}^1 \\
\vdots \\
\hline
\bar{\Gamma} \vdash x \vec{t} \vec{N} : \forall \beta (P_{s_n t_n}^n \rightarrow \mathbf{false}) \rightarrow \mathbf{false} \quad \bar{\Gamma} \vdash L : \forall \beta (P_{s_n t_n}^n \rightarrow \mathbf{false}) \\
\hline
\bar{\Gamma} \vdash (x \vec{t} \vec{N}) L : \mathbf{false}
\end{array}$$

For  $i \in \{1, \dots, n-1\}$  we can apply Lemma 36 to  $\bar{\Gamma} \vdash N_i : P_{s_i t_i}^i$  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  $\Gamma$  we still need  $\Gamma \vdash \forall \beta (P^n(s_n, t_n) \rightarrow \mathbf{false})$ .

To deduce this we have to take a closer look at  $L$ . First note that because either  $\alpha_n = \beta = s_n$  or there exists an  $i \in \{1, \dots, n-1\}$  such that  $\alpha_n \in \text{FV}(P^i(\alpha_i, \beta_i))$  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  $t_1, \dots, t_l, t' \in T_{\lambda_2}$  with  $M' := y t_1 \dots t_l$ , 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  $A = \forall \vec{\alpha} (P^n(s, t) \rightarrow \mathbf{false}) \in \Gamma$ . It is easy to see that in all three cases we can indeed deduce  $\Gamma \vdash \forall \beta (P^n(s_n, t_n) \rightarrow \mathbf{false})$ .

If  $L = \Lambda \beta. M_1 M_2$  for some  $M_1, M_2 \in \Lambda_{T_{\lambda_2}}$  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  $t'_1, \dots, t'_l \in T_{\lambda_2}$ ,  $n' \in \mathbb{N}$ , and some  $N'_1, \dots, N'_{n'} \in \Lambda_{T_{\lambda_2}}$ . We get that  $L = \Lambda \beta. x_A \vec{t}' \vec{N}'$  where  $\vec{N}' := N'_1 \dots N'_{n'} M_2$ . Hence,  $\beta \notin \text{FV}(\bar{\Gamma})$ , we can use the  $\forall$ -Introduction rule to deduce  $\bar{\Gamma} \vdash x_A \vec{t}' \vec{N}' : P_{s_n t_n}^n \rightarrow \mathbf{false}$ . Now we can conclude  $\Gamma \vdash P^n(s_n, t_n) \rightarrow \mathbf{false}$ .

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

If  $L = \Lambda\beta.\lambda y : t'.N$  for some  $y \in \mathcal{V}_V$ , some  $t' \in T_{\lambda_2}$ , and some  $N \in \Lambda_{T_{\lambda_2}}$  then  $t' = P^n_{s_n t_n}$ . Furthermore:

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

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

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

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

Now we can show that **false** is a semantic consequence of  $\Gamma$ .

$$\frac{\Gamma \vdash \forall\vec{\alpha}(P^1(\alpha_1, \beta_1) \rightarrow \dots \rightarrow \forall\beta(P^n(\alpha_n, \beta_n) \rightarrow \mathbf{false}) \rightarrow \mathbf{false})}{\frac{\Gamma \vdash P^1(s_1, t_1) \rightarrow \dots \rightarrow \forall\beta(P^n(s_n, t_n) \rightarrow \mathbf{false}) \rightarrow \mathbf{false} \quad \Gamma \vdash P^1(s_1, t_1)}{\vdots}}{\frac{\Gamma \vdash \forall\beta(P^n(s_n, t_n) \rightarrow \mathbf{false}) \rightarrow \mathbf{false} \quad \Gamma \vdash \forall\beta(P^n(s_n, t_n) \rightarrow \mathbf{false})}{\Gamma \vdash \mathbf{false}}}$$

$M = \lambda x : t_1.M'$  for some  $M' \in \Lambda_{T_{\lambda_2}}$ , some  $x \in \mathcal{V}_V \setminus \text{dom}(\Gamma)$ , and some  $t_1 \in T_{\lambda_2}$ . It follows that  $t = t_1 \rightarrow t_2$  for some  $t_2 \in T_{\lambda_2}$  which contradicts  $t = \mathbf{false}$ . So this case is impossible.

$M = \Lambda\gamma.M'$  for some  $M' \in \Lambda_{T_{\lambda_2}}$  and some  $\gamma \in \mathcal{V}_T$ .

It follows that  $t = \forall\gamma.t'$  for some  $t' \in T_{\lambda_2}$ . Again the fact that  $t = \mathbf{false}$  leads to a contradiction and makes this case impossible.

$M = M' t'$  for some  $M' \in \Lambda_{T_{\lambda_2}}$  and some  $t' \in T_{\lambda_2}$ .

Since  $M$  is in normal form we have that  $M' = x M_1 \dots M_n$  for some  $x \in \mathcal{V}_V$ ,  $n \in \mathbb{N}$ ,



and some  $M_1, \dots, M_n \in \Lambda_{T_{\lambda 2}} \cup T_{\lambda 2}$ . 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. □

**Lemma 38.**

$$\Gamma \vdash \mathbf{false} \quad \text{iff} \quad \text{There is a } \lambda 2 \text{ term } M \text{ such that } \overline{\Gamma} \vdash M : \mathbf{false}$$

*Proof.* The  $\Leftarrow$  direction follows from Lemma 37. And the  $\Rightarrow$  direction follows from Lemma 35 with  $A = \mathbf{false}$ . □

**Theorem 39.** *INHAB is undecidable.*

*Proof.* From Lemma 38 it follows that  $\mathbf{CONS} \leq \mathbf{INHAB}$ . Since, by Theorem 33,  $\mathbf{CONS}$  is undecidable we have shown that  $\mathbf{INHAB}$  is undecidable too. □

## References

- [1] H.P. Barendregt, 1993. Lambda calculi with types, Handbook of Logic in Computer Science, Volume II, 34-68.
- [2] P. Urzyczyn, 1997. Inhabitation in Typed Lambda-Calculi (A Syntactic Approach), Typed Lambda Calculi and Applications.