値依存時相仕様検証のための型システム

南條 陽史

プログラムの形式検証

プログラム

形式仕様

$$p \stackrel{?}{\models} \phi$$

プログラムpが形式仕様ϕを満たすか？

本研究：値依存時相仕様検証

関数型プログラム

値依存時相仕様

$$p \stackrel{?}{\models} \phi$$

プログラムpが形式仕様ϕを満たすか？

本研究：値依存時相仕様検証

関数型プログラム

値依存時相仕様

$$p \stackrel{?}{\models} \phi$$

プログラムpが形式仕様ϕを満たすか？

関数型プログラム

数学的な関数を組み合わせて計算を扱うスタイルのプログラム

例)

実行するとSendイベントを発行

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs (n-1)
```

生成されうるイベント列

$n < 0$: Send, Send, .. (無限列)

$n = 0 : \epsilon$

$n = 1$: Send

$n = 2$: Send, Send

…

本研究：値依存時相仕様検証

関数型プログラム

値依存時相仕様

$$p \stackrel{?}{\models} \phi$$

プログラムpが形式仕様ϕを満たすか？

本研究：値依存時相仕様検証

有限のイベント列に対する仕様

無限のイベント列に対する仕様

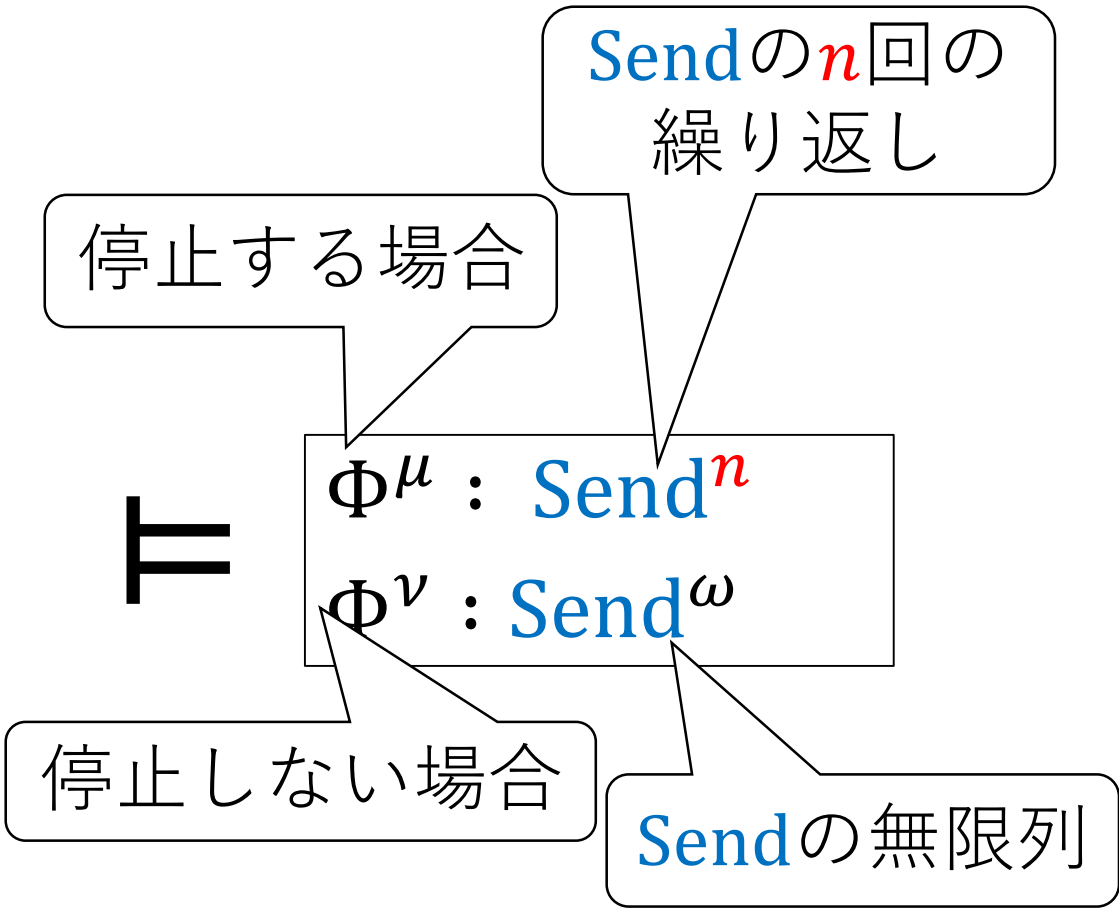
$$p \stackrel{?}{\models} (\Phi^{\mu}, \Phi^{\nu})$$

プログラムpの生成する有限のイベント列がΦ^{μ}を満たし無限のイベント列がΦ^{ν}を満たす

値依存時相仕様

プログラムが生成するイベント列に対する実行時の値に依存した仕様

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs (n-1)
```



その他の検証可能な仕様例：ならし計算量

以下のプログラムが生成する有限イベント列xに対する仕様:

$$\#_{Enq}(x) = \#_{Tick}(x) = \#_{Deq}(x)$$

```
let rev l =
  let rec aux l acc = match l with
    | [] -> acc | h::t ->
      event[Tick]; aux t (h::acc)
  in aux l []
let is_empty (l1,l2) = l1 = [] && l2 = []
let enqueue e (l1,l2)
=event[Enq];(l1,e::l2)
let rec dequeue (l1,l2) = match l1 with
  | [] -> dequeue (rev l2, [])
  | e::l1' -> event[Deq]; (e, (l1', l2))
let rec main (l1,l2) =
  if * then main (enqueue 42 (l1,l2))
  else if is_empty (l1,l2) then ()
  else main (snd (dequeue (l1,l2)))
```

$\#_a(x)$ … イベント列xに現れるイベントaの数

プログラムが停止して得られる有限イベント列の例：

Enq, Enq, Tick, Tick, Deq, Deq

Enq, Tick, Deq, Enq, Tick, Deq

Enq, Enq, Tick, Tick, Deq, Deq, Enq, Tick, Deq

本研究：値依存時相仕様

関数型プログラム

イベント列に対する仕様

$$p \stackrel{?}{\models} (\Phi^{\mu}, \Phi^{\nu})$$

プログラムpの生成する有限のイベント列がΦ^{μ}を満たし無限のイベント列がΦ^{ν}を満たす

提案する検証手法

pが生成する有限イベント列の述語表現

pが生成する無限イベント列の述語表現

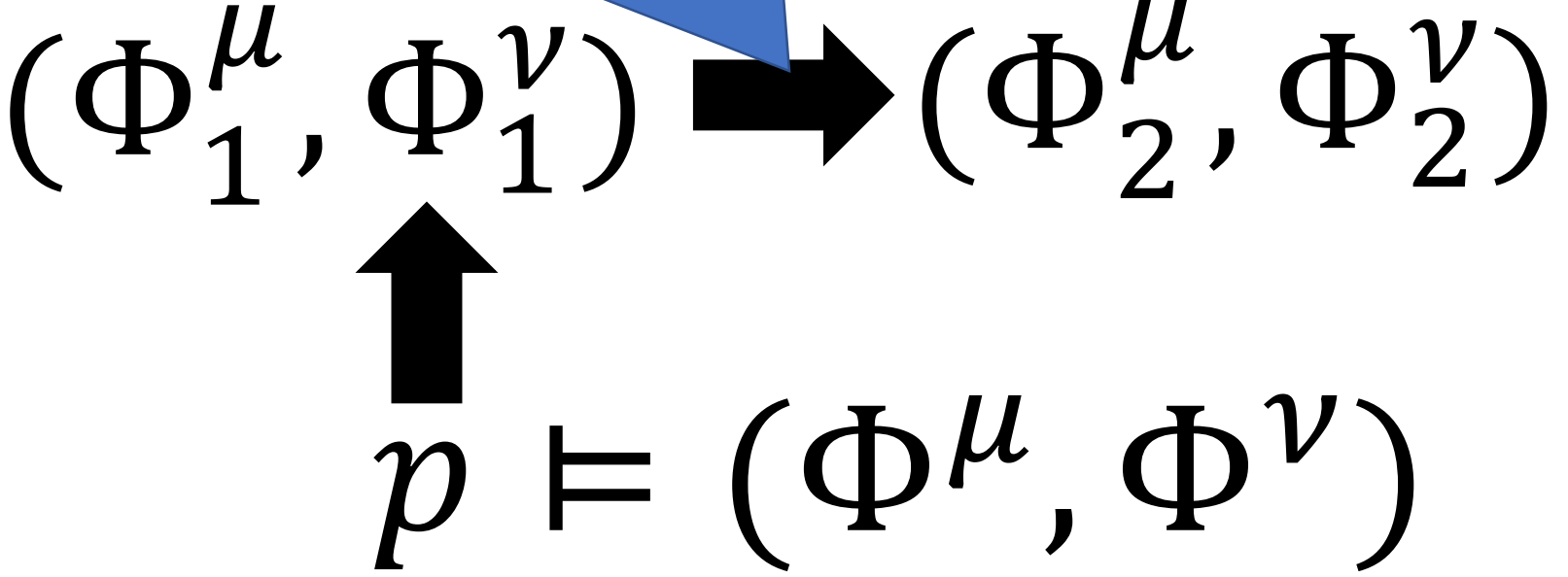
$(\Phi_1^{\mu}, \Phi_1^{\nu})$

型システム

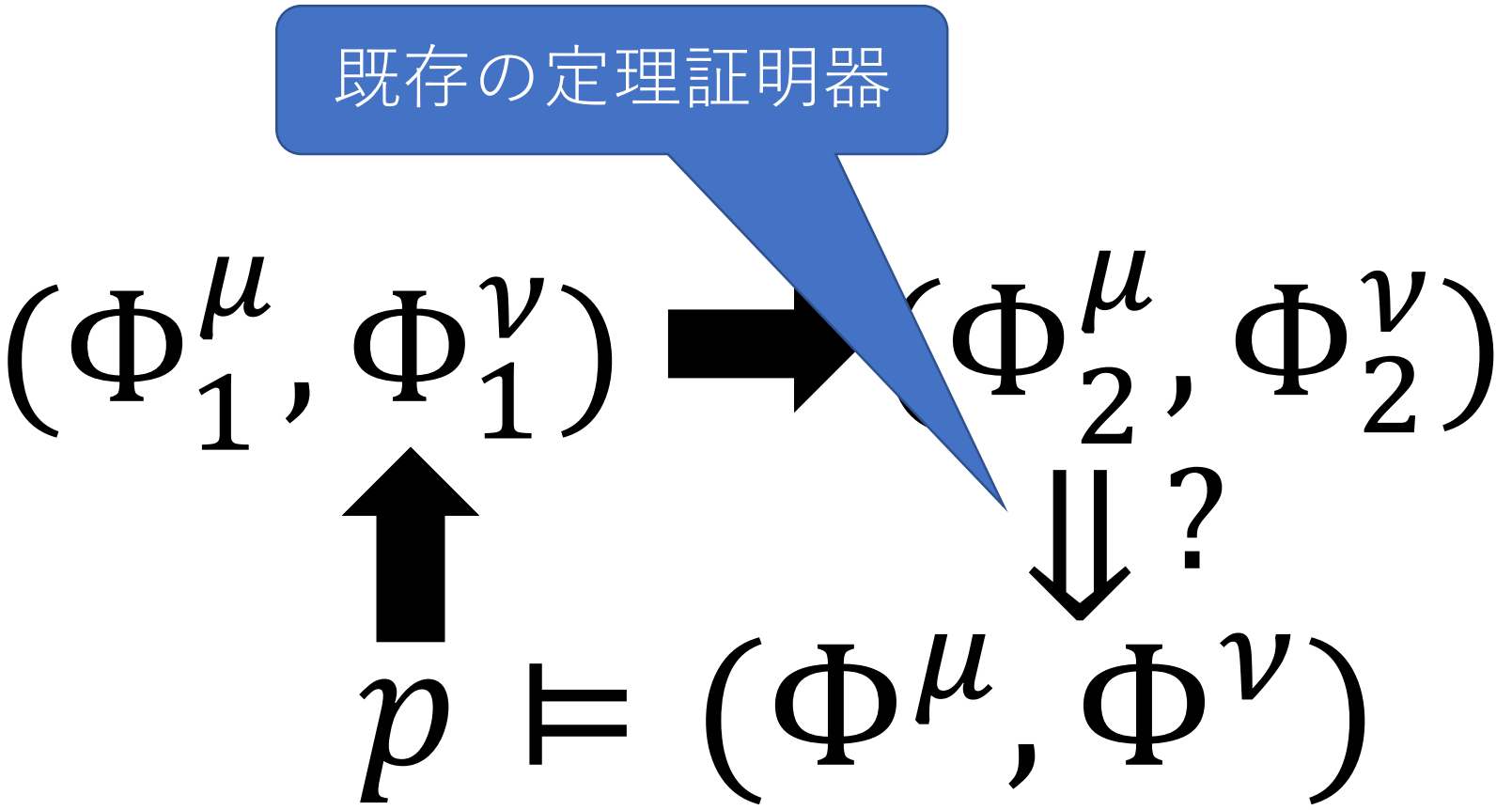
$p \models (\Phi^{\mu}, \Phi^{\nu})$

提案する検証手法

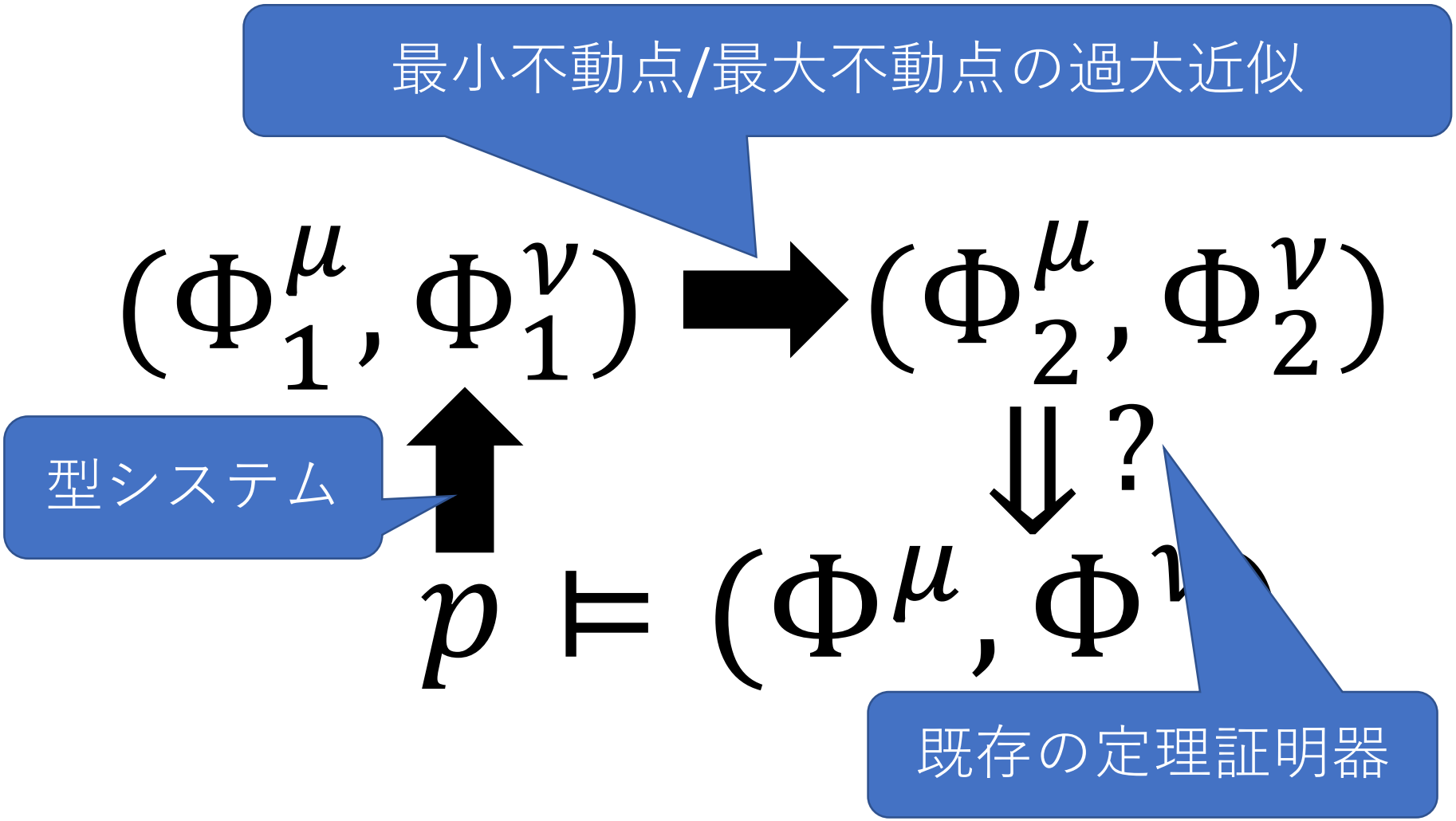
最小不動点/最大不動点の過大近似



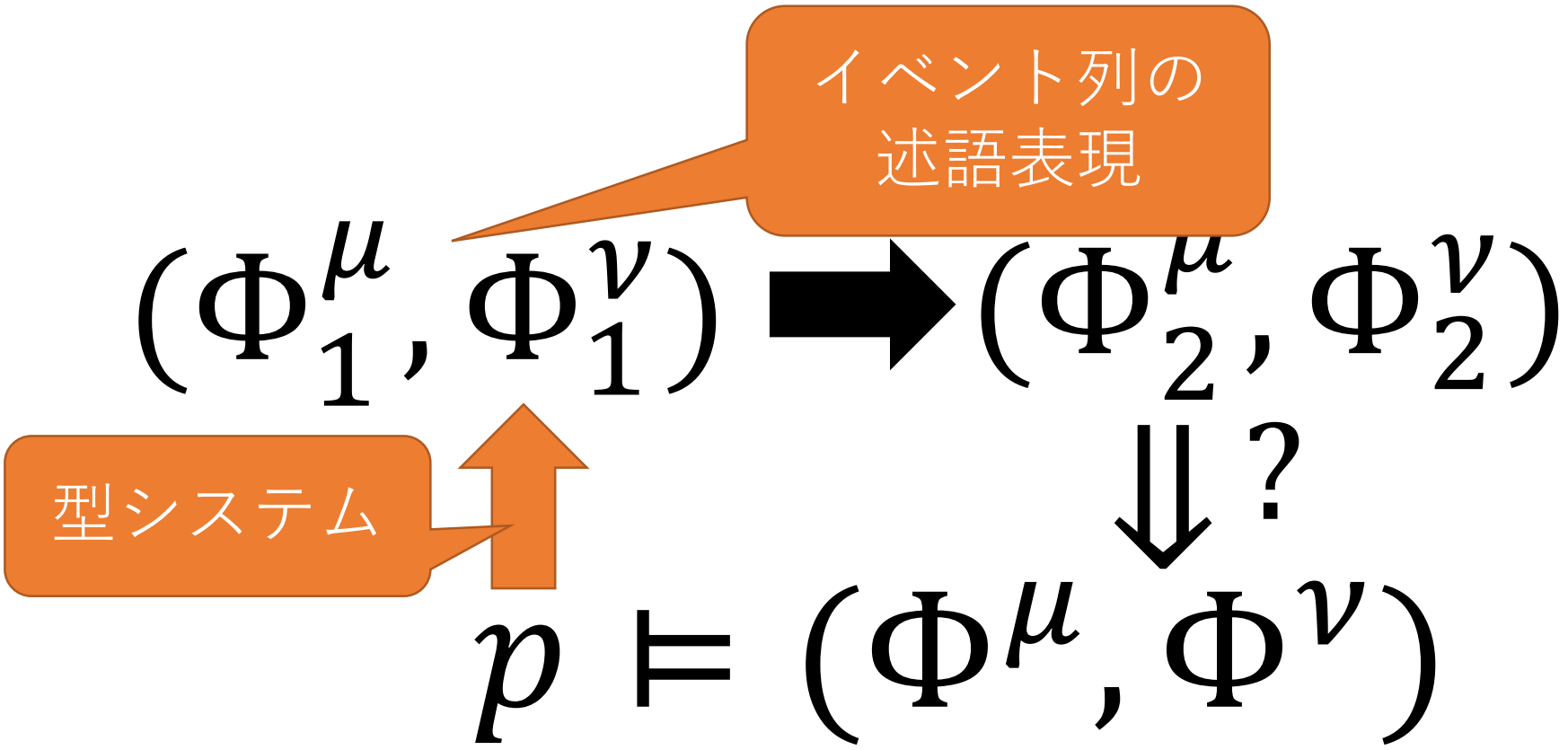
提案する検証手法


既存の定理証明器
$(\Phi_1^{\mu}, \Phi_1^{\nu})$
$(\Phi_2^{\mu}, \Phi_2^{\nu})$
$\Downarrow ?$
$p \models (\Phi^{\mu}, \Phi^{\nu})$

提案する検証手法


最小不動点/最大不動点の過大近似
$(\Phi_1^{\mu}, \Phi_1^{\nu})$
$(\Phi_2^{\mu}, \Phi_2^{\nu})$
型システム
$\Downarrow ?$
$p \vDash (\Phi^{\mu}, \Phi^{\nu})$
既存の定理証明器

提案する検証手法


イベント列の
述語表現
$(\Phi_1^{\mu}, \Phi_1^{\nu})$
$(\Phi_2^{\mu}, \Phi_2^{\nu})$
型システム
$\Downarrow ?$
$p \models (\Phi^{\mu}, \Phi^{\nu})$

有限イベント列の解析例

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

有限イベント列の解析例

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

send_msgsの有限イベント列xが$X(n, x)$を満たすとする
(ただしXは述語変数)

有限イベント列の解析例

send_msgsの有限イベント列xが$X(n,x)$を満たすとする
(ただしXは述語変数)

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

$X(n-1,x)$

有限イベント列の解析例

send_msgsの有限イベント列xが$X(n, x)$を満たすとする
(ただしXは述語変数)

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

$\exists y.\, x = \mathrm{Send} \cdot y \wedge X(n-1, y)$

$X(n-1, x)$

有限イベント列の解析例

send_msgsの有限イベント列xが$X(n, x)$を満たすとする
(ただしXは述語変数)

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

$x = \epsilon$

$\exists y.\, x = \mathrm{Send} \cdot y \wedge X(n-1, y)$

$X(n-1, x)$

有限イベント列の解析例

send_msgsの有限イベント列xが$X(n,x)$を満たすとする
(ただしXは述語変数)

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

$x = \epsilon$

$\exists y.\, x = \mathrm{Send} \cdot y \wedge X(n-1, y)$

$X(n-1, x)$

$$n = 0 \wedge x = \epsilon \vee n \neq 0 \wedge (\exists y.\, x = \mathrm{Send} \cdot y \wedge X(n-1, y))$$

有限イベント列の解析例

send_msgsの有限イベント列xが$X(n,x)$を満たすとする
(ただしXは述語変数)

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

述語変数Xの満たすべき制約

$$X(n,x) \equiv \begin{pmatrix} n = 0 \wedge x = \epsilon \vee \\ n \neq 0 \wedge (\exists y.\, x = \mathrm{Send} \cdot y \wedge X(n-1, y)) \end{pmatrix}$$

有限イベント列の解析例

```
func send_msgs(n)
  if n = 0 then
    ()
  else
    event[Send];
    send_msgs(n-1)
```

send_msgsの有限イベント列xが$X(n,x)$を満たすとする
(ただしXは述語変数)

Xの制約を満たす解を表す最小不動点

$$(\mu X(n,x).\begin{pmatrix} n=0 \wedge x=\epsilon \vee \\ n\neq 0 \wedge (\exists y.\, x = \mathrm{Send}\cdot y \wedge X(n-1,y)) \end{pmatrix})(x,n)$$

提案する型システム

型判断式の導出規則

$$\frac{}{\Gamma \vdash n\colon (\{x \mid x = n\} \,\&\, \Phi_{val})}\ \text{T-Const}$$

$$\frac{sty(\Gamma(x)) = \texttt{int}}{\Gamma \vdash x\colon (\{u \mid u = x\} \,\&\, \Phi_{val})}\ \text{T-VInt} \qquad \frac{sty(\Gamma(x)) \neq \texttt{int}}{\Gamma \vdash x\colon (\Gamma(x) \,\&\, \Phi_{val})}\ \text{T-VFun}$$

$$\frac{x \notin fv(\tau_2) \cup fv(\Phi_2) \quad \Gamma \vdash e_1\colon (\tau_1 \,\&\, \Phi_1) \quad \Gamma, x : \tau_1 \vdash e_2\colon (\tau_2 \,\&\, \Phi_2)}{\Gamma \vdash \texttt{let}\ x = e_1\ \texttt{in}\ e_2\colon (\tau_2 \,\&\, \Phi_1 \cdot \Phi_2)}\ \text{T-Let}$$

$$\frac{\Gamma \vdash v_1\colon ((x : \tau) \to (\tau' \,\&\, \Phi) \,\&\, \Phi_{val}) \quad \Gamma \vdash v_2\colon (\tau \,\&\, \Phi_{val})}{\Gamma \vdash v_1\ v_2\colon [v_2/x](\tau' \,\&\, \Phi)}\ \text{T-App}$$

$$\frac{\Gamma \vdash v_1\colon (\texttt{int} \,\&\, \Phi_{val}) \quad \Gamma \vdash v_2\colon (\texttt{int} \,\&\, \Phi_{val})}{\Gamma \vdash v_1\ op\ v_2\colon (\{x \mid x = v_1\ op\ v_2\} \,\&\, \Phi_{val})}\ \text{T-Op}$$

$$\frac{\Gamma, v = 0 \vdash e_1\colon \sigma \quad \Gamma, v \neq 0 \vdash e_2\colon \sigma}{\Gamma \vdash \texttt{ifz}\ v\ \texttt{then}\ e_1\ \texttt{else}\ e_2\colon \sigma}\ \text{T-If}$$

$$\frac{\Phi = (\lambda x \in \Sigma^*.x = \underline{\mathbf{a}}, \lambda x \in \Sigma^\omega.\bot)}{\Gamma \vdash \texttt{ev}[\underline{\mathbf{a}}]\colon (\{x \mid x = 0\} \,\&\, \Phi)}\ \text{T-Event}$$

$$\frac{\begin{array}{c}\tau'_f = (\widetilde{x}\colon \widetilde{\tau}) \to (\tau \,\&\, (\lambda x \in \Sigma^*.X_\mu(\widetilde{x}, x), \lambda x \in \Sigma^\omega.X_\nu(\widetilde{x}, x))) \\ \Gamma, f : \tau'_f, \widetilde{x} : \widetilde{\tau} \vdash e\colon (\tau \,\&\, \Phi) \\ q_\mu = \mu X_\mu(\widetilde{x}, x).\Phi^\mu(x) \quad q_\nu = \nu X_\nu(\widetilde{x}, x).[q_\mu/X_\mu]\Phi^\nu(x) \\ \tau_f = (\widetilde{x}\colon \widetilde{\tau}) \to (\tau \,\&\, (\lambda x \in \Sigma^*.q_\mu(\widetilde{x}, x), \lambda x \in \Sigma^\omega.q_\nu(\widetilde{x}, x)))\end{array}}{\Gamma \vdash \texttt{rec}(f, \widetilde{x}, e)\colon (\tau_f \,\&\, \Phi_{val})}\ \text{T-Fun}$$

$$\frac{\Gamma \vdash e\colon \sigma_1 \quad \Gamma \vdash \sigma_1 <: \sigma_2}{\Gamma \vdash e\colon \sigma_2}\ \text{T-Sub}$$

型判断式 $\vdash e : (\tau \,\&\, (\Phi^\mu, \Phi^\nu))$

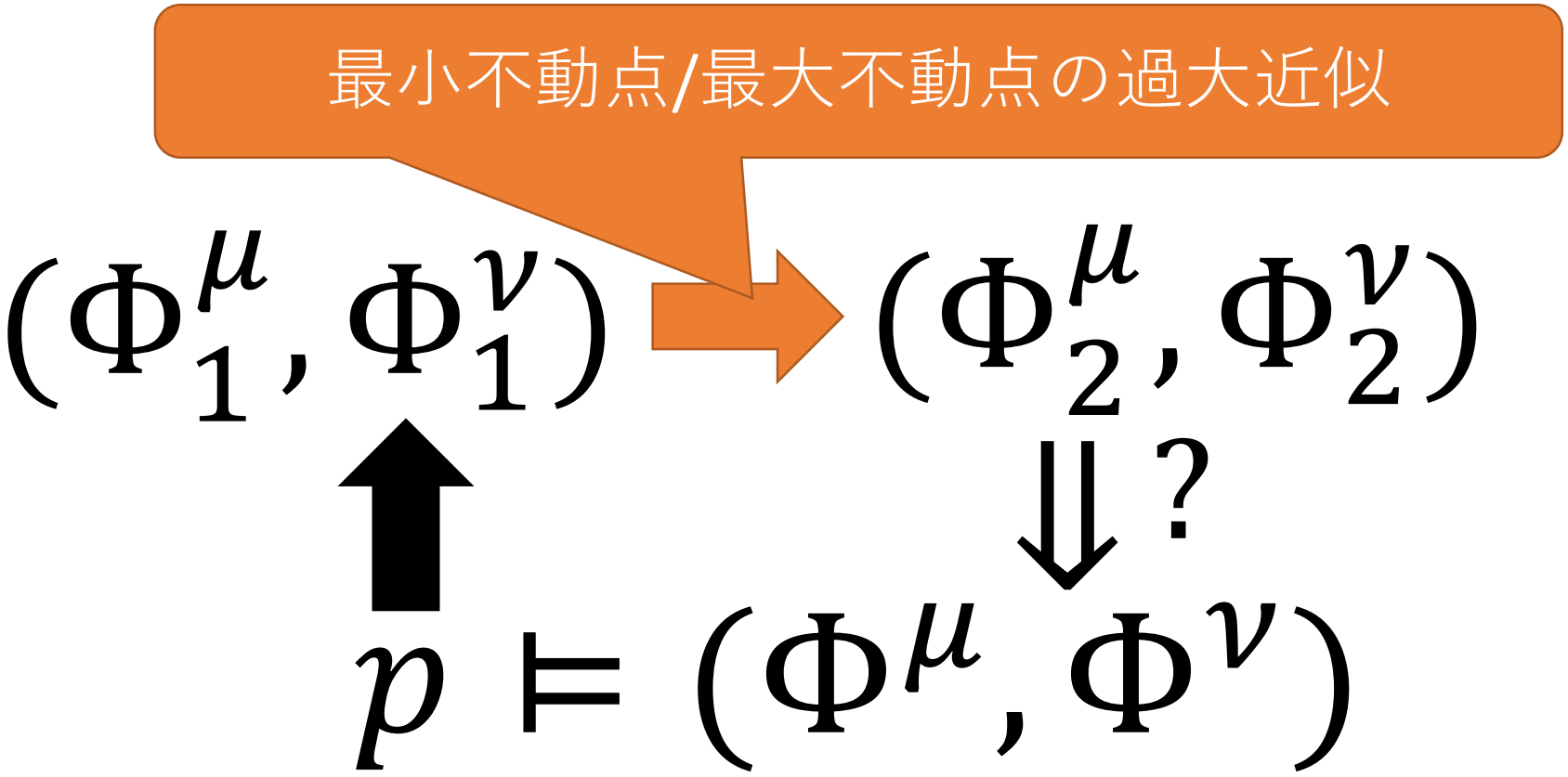
プログラム

値に対する制約

イベント列に対する述語

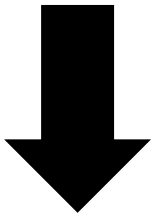
健全性：型判断式が成り立つとeの実行時の挙動はそれぞれの制約を満たす

提案する検証手法


最小不動点/最大不動点の過大近似
$(\Phi_1^{\mu}, \Phi_1^{\nu})$
$(\Phi_2^{\mu}, \Phi_2^{\nu})$
$\Downarrow ?$
$p \models (\Phi^{\mu}, \Phi^{\nu})$

最小不動点の過大近似例

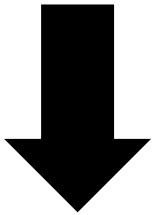
$$\Phi_1^{\mu} = \left(\mu X(n,x).\ \begin{array}{c} n = 0 \wedge x = \epsilon \vee \\ n \neq 0 \wedge \left(\exists y.\ x = \mathrm{Send} \cdot y \wedge X(n-1,y)\right) \end{array}\right)(n,x)$$



$$\Phi_2^{\mu} = \ (n \geq 0 \wedge x = \mathrm{Send}^n)$$

最大不動点の過大近似例

$$\Phi_1^{\nu} =$$
$$(\nu X(n,x).\, n \neq 0 \wedge \left(\exists y.\ x = \mathrm{Send} \cdot y \wedge X(n-1,y)\right))(n,x)$$



$$\Phi_2^{\nu} = \quad (n < 0 \wedge x = \mathrm{Send}^{\omega})$$

提案する不動点近似法

導出規則

$$\frac{\models p_1(\overline{x}) \wedge \psi' \Rightarrow \psi}{X(\overline{x}); p_1; p_2; \psi' \downarrow \psi}\ \text{Apx}^{\mu}\text{-Base}$$

$$\frac{\models p_1(\overline{x}) \wedge \psi \Rightarrow p_1(\overline{t}) \wedge p_2(\overline{x},\overline{t})}{X(\overline{x}); p_1; p_2; \psi \downarrow X(\overline{t})}\ \text{Apx}^{\mu}\text{-Rec}$$

$$\frac{X(\overline{x}); p_1; p_2; \psi \downarrow \psi_1 \quad X(\overline{x}); p_1; p_2; \psi \downarrow \psi_2}{X(\overline{x}); p_1; p_2; \psi \downarrow \psi_1 \wedge \psi_2}\ \text{Apx}^{\mu}\text{-}\wedge$$

$$\frac{\begin{array}{c}\models (p_1(\overline{x}) \wedge \psi) \Rightarrow (\psi_1' \vee \psi_2') \quad fv(\psi_i') \subseteq \{\overline{x}\} \quad X \notin fpv(\psi_i') \\ X(\overline{x}); p_1; p_2; \psi \wedge \psi_i' \downarrow \psi_i \quad (i = 1,2)\end{array}}{X(\overline{x}); p_1; p_2; \psi \downarrow \psi_1 \vee \psi_2}\ \text{Apx}^{\mu}\text{-}\vee$$

$$\frac{\begin{array}{c}X(\overline{x}); p_1; p_2; \psi' \downarrow [x'/x]\psi \\ x' \notin fv(\psi') \cup fv(\psi) \cup \{\overline{x}\} \cup fv(p_1) \cup fv(p_2)\end{array}}{X(\overline{x}); p_1; p_2; \psi' \downarrow \forall x.\psi}\ \text{Apx}^{\mu}\text{-}\forall$$

$$\frac{\begin{array}{c}\models (p_1(\overline{x}) \wedge \psi') \Rightarrow \exists x'.\psi'' \quad fv(\psi'') \subseteq \{\overline{x}\} \quad X \notin fpv(\psi'') \\ X(\overline{x}); p_1; p_2; \psi' \wedge \psi'' \downarrow [x'/x]\psi \\ x' \notin fv(\psi') \cup fv(\psi) \cup \{\overline{x}\} \cup fv(p_1) \cup fv(p_2)\end{array}}{X(\overline{x}); p_1; p_2; \psi' \downarrow \exists x.\psi}\ \text{Apx}^{\mu}\text{-}\exists$$

$$\frac{\models p_1(\overline{x}) \wedge \psi' \Rightarrow \neg\psi}{X(\overline{x}); p_1; p_2; \psi' \,\overline{\uparrow}\, \psi}\ \text{Apx}^{\nu}\text{-Base}$$

$$\frac{\models p_1(\overline{x}) \wedge \psi \Rightarrow p_1(\overline{t}) \wedge p_2(\overline{x},\overline{t})}{X(\overline{x}); p_1; p_2; \psi \,\overline{\uparrow}\, X(\overline{t})}\ \text{Apx}^{\nu}\text{-Rec}$$

$$\frac{\begin{array}{c}\models (p_1(\overline{x}) \wedge \psi) \Rightarrow (\psi_1' \vee \psi_2') \quad fv(\psi_i') \subseteq \{\overline{x}\} \quad X \notin fpv(\psi_i') \\ X(\overline{x}); p_1; p_2; \psi \wedge \psi_i' \,\overline{\uparrow}\, \psi_i \quad (i = 1,2)\end{array}}{X(\overline{x}); p_1; p_2; \psi \,\overline{\uparrow}\, \psi_1 \wedge \psi_2}\ \text{Apx}^{\nu}\text{-}\wedge$$

$$\frac{X(\overline{x}); p_1; p_2; \psi \,\overline{\uparrow}\, \psi_1 \quad X(\overline{x}); p_1; p_2; \psi \,\overline{\uparrow}\, \psi_2}{X(\overline{x}); p_1; p_2; \psi \,\overline{\uparrow}\, \psi_1 \vee \psi_2}\ \text{Apx}^{\nu}\text{-}\vee$$

$$\frac{\begin{array}{c}\models (p_1(\overline{x}) \wedge \psi') \Rightarrow \exists x'.\psi'' \quad fv(\psi'') \subseteq \{\overline{x}\} \quad X \notin fpv(\psi'') \\ X(\overline{x}); p_1; p_2; \psi' \wedge \psi'' \,\overline{\uparrow}\, [x'/x]\psi \\ x' \notin fv(\psi') \cup fv(\psi) \cup \{\overline{x}\} \cup fv(p_1) \cup fv(p_2)\end{array}}{X(\overline{x}); p_1; p_2; \psi' \,\overline{\uparrow}\, \forall x.\psi}\ \text{Apx}^{\nu}\text{-}\forall$$

$$\frac{\begin{array}{c}X(\overline{x}); p_1; p_2; \psi' \,\overline{\uparrow}\, [x'/x]\psi \\ x' \notin fv(\psi') \cup fv(\psi) \cup \{\overline{x}\} \cup fv(p_1) \cup fv(p_2)\end{array}}{X(\overline{x}); p_1; p_2; \psi' \,\overline{\uparrow}\, \exists x.\psi}\ \text{Apx}^{\nu}\text{-}\exists$$

健全性：

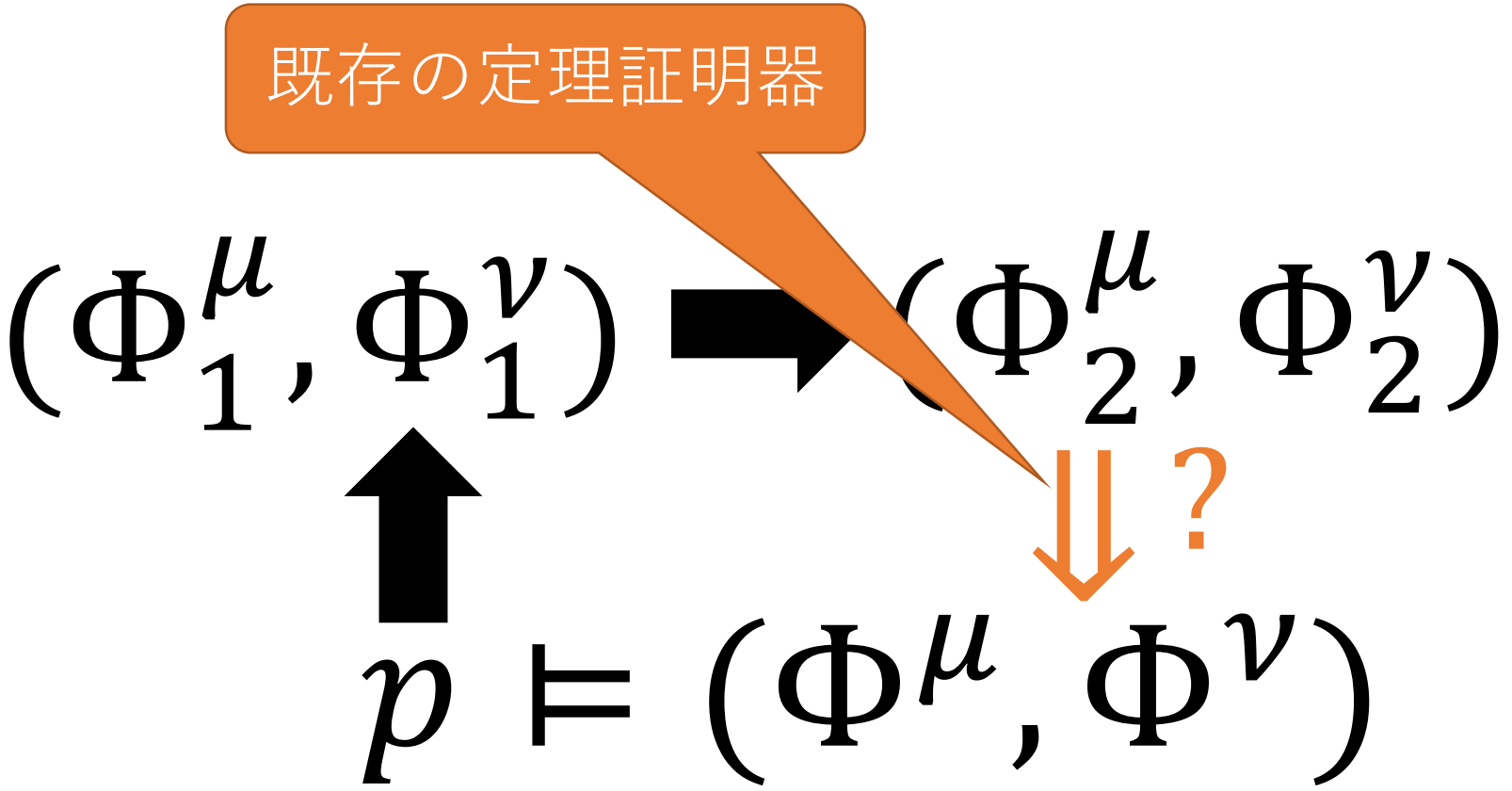
$X(\tilde{x}); p_1; p_2; \psi' \downarrow \psi$が成り立つならば

$$p_1(\tilde{x}) \Rightarrow (\mu X(\tilde{x}).\, \neg\psi' \vee \psi)(\tilde{x})$$

$X(\tilde{x}); p_1; p_2; \psi' \,\overline{\uparrow}\, \psi$が成り立つならば

$$(\nu X(\tilde{x}).\, \psi' \wedge \psi)(\tilde{x}) \Rightarrow \neg p_1(\tilde{x})$$

提案する検証手法



既存の定理証明器による含意関係判定

$$(n \geq 0 \land x = \mathrm{Send}^{n}) \stackrel{?}{\Rightarrow} x = \mathrm{Send}^{n}$$

$$(n < 0 \land x = \mathrm{Send}^{\omega}) \stackrel{?}{\Rightarrow} x = \mathrm{Send}^{\omega}$$

既存の定理証明器による含意関係判定

判定成功!

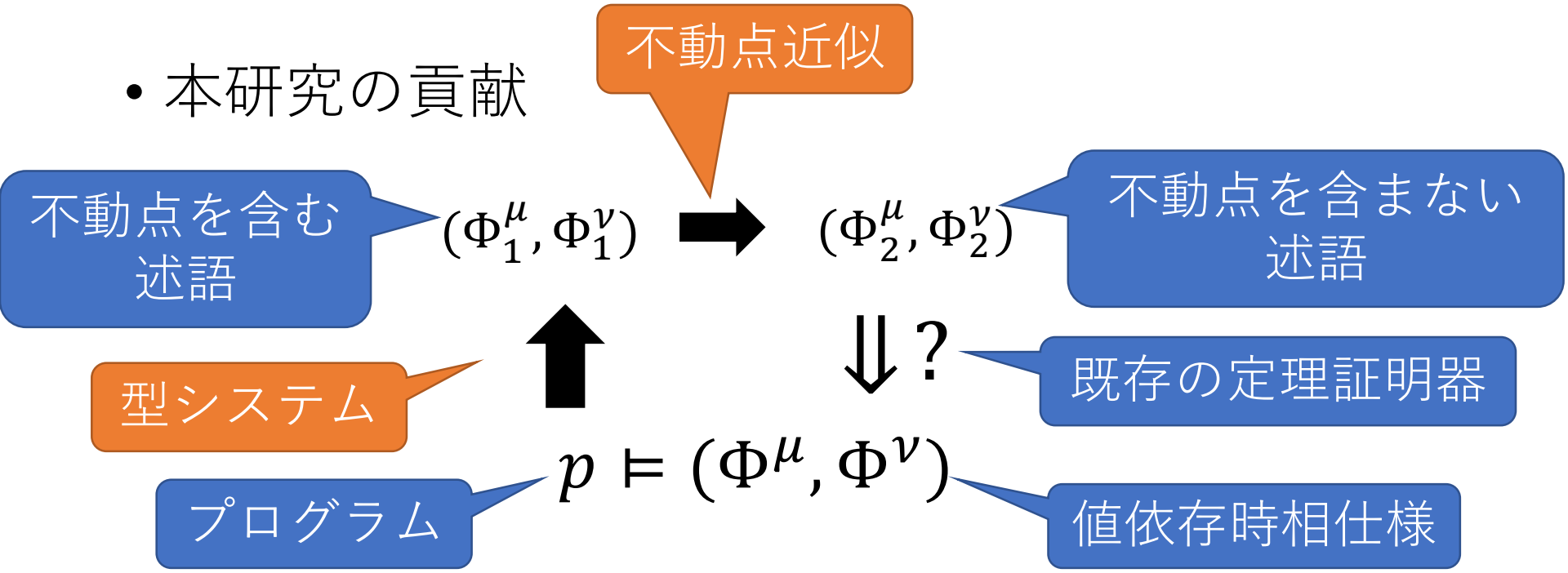
$$(n \geq 0 \wedge x = \mathrm{Send}^{n}) \Rightarrow x = \mathrm{Send}^{n}$$

判定成功!

$$(n < 0 \wedge x = \mathrm{Send}^{\omega}) \Rightarrow x = \mathrm{Send}^{\omega}$$

本研究の貢献と将来の課題

- 本研究の貢献



- 将来の課題
  - 仕様の表現力を広げる
  - 型検査器, 型推論器の実装