ソフトウェア基礎論配布資料 (8)
π 計算

五十嵐 淳
京都大学 大学院情報学研究科知能情報学専攻
e-mail: `igarashi@kuis.kyoto-u.ac.jp`

平成 19 年 1 月 23 日

1 π 計算—特徴

mobile process の計算体系:

- 並列実行される**プロセス**(*process*) によるシステム記述
- 主要な計算ステップ: **通信路/チャネル**(*communication channel*) を介してのプロセス間通信
- 主要なデータ: 通信チャネルの名前 (cf. URL)
- 新しい名前の生成機構と動的に変化する名前の有効範囲による通信トポロジーの動的な変化

1.1 概要

構文:

- $\mathbf{0}$ ⋯ 何もしない・実行終了状態にあるプロセス
- $P_1 \parallel P_2$ ⋯ プロセス P_1 と P_2 の並列実行
- $\tau.P$ ⋯ (観測できない) 内部動作を行った後，P を実行
- $x![y].P$ ⋯ 通信チャネル x へ y を送信，送信後に P を実行
- $x?[z].P$ ⋯ 通信チャネル x からデータを受信，パラメータ z (有効範囲は P) をそれに束縛して P を実行

- $(\boldsymbol{\nu}x)P$ ⋯ 新しい通信チャネルを生成し，その名前を x として P を実行
- $*P$ ⋯ P の無限コピー ($P \parallel P \parallel \cdots \parallel P \parallel \cdots$)
- $P_1 + P_2$ ⋯ プロセス P_1 か P_2 の，可能な動作のうち一方を実行 (両方可能でも，どちらか一方を非決定的に選択)

なお，並列実行は $P \mid Q$，$x![y].P$ は $\bar{x}\langle y\rangle.P$，$x?[y].P$ は $x(y).P$, $*P$ は $!P$ と表記されることも (オリジナルの文献も含め) 多い．

主要な計算規則:

$$x![y].P \parallel x?[z].Q \longrightarrow P \parallel [y/z]Q$$

($[y/z]$ は z を y で変数参照関係を保ちながら置き換える操作．) 非決定性:

$$\begin{array}{ccc} x![y].P_1 \parallel x![w].P_2 \parallel x?[z].P_3 & \longrightarrow & P_1 \parallel x![w].P_2 \parallel [y/z]P_3 \\ \downarrow & & \\ x![y].P_1 \parallel P_2 \parallel [w/z]P_3 & & \end{array}$$

名前の生成 $((\boldsymbol{\nu}x)P)$ **について:**

$$\begin{array}{rcl} ((\boldsymbol{\nu}x)x![y].P) \parallel x?[z].Q & \not\longrightarrow & (\boldsymbol{\nu}x)P \parallel [y/z]Q \\ ((\boldsymbol{\nu}x)\#^1 x![y].P) \parallel x?[z].Q & \longrightarrow & ((\boldsymbol{\nu}x)P) \parallel [y/z]Q \end{array}$$

また，ν で束縛された名前が送受信対象になると有効範囲の移動が発生する．P に y が現われないとすると，

$$\begin{array}{rcl} ((\boldsymbol{\nu}y)x![y].P \parallel y?[w].Q) \parallel x?[z].z![v].R & \longrightarrow & P \parallel ((\boldsymbol{\nu}y)y?[w].Q \parallel y![v].R') \\ & \longrightarrow & P \parallel ((\boldsymbol{\nu}y)[v/w]Q \parallel R') \end{array}$$

1.2 例

「1 足す」サーバ: s でリクエストを受信して r に返事を送信.

$$\begin{array}{l} P \equiv *s?[n].r![n+1].\mathbf{0} \\ P \parallel s![2].r?[x].Q \longrightarrow \cdots \longrightarrow ? \\ P \parallel s![2].r?[x].Q \parallel s![4].r?[y].R \longrightarrow \cdots \longrightarrow ? \end{array}$$

「1 足す」サーバ (改良版)

$$\begin{array}{l} P \equiv *s?[n,r].r![n+1].\mathbf{0} \\ P \parallel ((\boldsymbol{\nu}r')s![2,r'].r'?[x].Q) \longrightarrow \cdots \longrightarrow ?? \\ P \parallel ((\boldsymbol{\nu}r_1)s![2,r_1].r_1?[x].Q) \parallel ((\boldsymbol{\nu}r_2)s![4,r_2].r_2?[y].R) \longrightarrow \cdots \longrightarrow ?? \end{array}$$

点オブジェクト:

$$\begin{array}{rcl} P & \equiv & (\boldsymbol{\nu} state) \\ & & \quad state![0]. \parallel \\ & & \quad *state?[n].(get?[r].r![n].state![n].\mathbf{0} + set?[n', r].r![].state![n'].\mathbf{0}) \\ \multicolumn{3}{l}{P \parallel ((\boldsymbol{\nu} r_1)get![r_1].r_1?[n].Q_1) \parallel ((\boldsymbol{\nu} r_2)set![3, r_2].r_2?[].Q_2)} \end{array}$$

名前の組の通信の単一名通信による模倣:

$$\begin{array}{rcl} x![y_1, \ldots, y_n].P & \equiv & (\boldsymbol{\nu} w)x![w].w![y_1].\cdots.w![y_n].P \\ x?[z_1, \ldots, z_n].Q & \equiv & x?[v].v?[z_1].\cdots.v?[z_n].Q \end{array}$$

2　π 計算の形式的定義

2.1　構文

環境 Γ は変数 (宣言) の並びとする．判断 $\Gamma \vdash \#^i x \in \mathbf{Ch}$, $\Gamma \vdash P \in \mathbf{Pr}$, $\Gamma \vdash G \in \mathbf{GPr}$ はそれぞれ「$\#^i x$ が Γ のもとで有効な変数参照である」「Γ のもとで P はプロセスである」「Γ のもとで G は動作ガード付きプロセスである」ことを表し，以下の規則で定義される．

$$\Gamma, x \vdash \#^0 x \in \mathbf{Ch} \qquad (\textsc{VarRef0})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Ch}}{\Gamma, x \vdash \#^{i+1} x \in \mathbf{Ch}} \qquad (\textsc{VarRef1})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Ch} \qquad (x \not\equiv y)}{\Gamma, y \vdash \#^i x \in \mathbf{Ch}} \qquad (\textsc{VarRef2})$$

$$\frac{\Gamma \vdash P_1 \in \mathbf{Pr} \qquad \Gamma \vdash P_2 \in \mathbf{Pr}}{\Gamma \vdash P_1 \parallel P_2 \in \mathbf{Pr}} \qquad (\text{P-}\textsc{Par})$$

$$\frac{\Gamma, x \vdash P \in \mathbf{Pr}}{\Gamma \vdash (\boldsymbol{\nu} x)P \in \mathbf{Pr}} \qquad (\text{P-}\textsc{New})$$

$$\frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash *P \in \mathbf{Pr}} \qquad (\text{P-}\textsc{Rep})$$

$$\frac{\Gamma \vdash P \in \mathbf{GPr}}{\Gamma \vdash P \in \mathbf{Pr}} \qquad (\text{P-GP}\textsc{r})$$

$$\Gamma \vdash \mathbf{0} \in \mathbf{GPr} \qquad (\text{G-}\textsc{Nil})$$

$$\frac{\Gamma \vdash G_1 \in \mathbf{GPr} \qquad \Gamma \vdash G_2 \in \mathbf{GPr}}{\Gamma \vdash G_1 + G_2 \in \mathbf{GPr}} \qquad (\text{G-}\textsc{Choice})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Ch} \qquad \Gamma \vdash \#^j y \in \mathbf{Ch} \qquad \Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash \#^i x![\#^j y].P \in \mathbf{GPr}} \qquad (\text{G-}\textsc{Out})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Ch} \qquad \Gamma, z \vdash P \in \mathbf{Pr}}{\#^i x?[z].P \in \mathbf{GPr}} \qquad (\text{G-}\textsc{In})$$

$$\frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash \tau.P \in \mathbf{GPr}} \qquad (\text{G-}\textsc{Tau})$$

選択的動作を行う $+$ で結合されるのは，$\mathbf{0}$ と 入出力・内部動作を行うプロセスに限られており，$(P \parallel Q) + R$ のような構成は許されていない．

2.2 簡約

概要で見たように，同じ名前のチャネルに対する入出力の通信が計算ステップとしてモデル化されているが，その定義は λ 計算にはない困難さが伴なう．その理由は，

- 通信を起こすプロセスが，必ずしも項の構造の中で近くにあるわけではない．

$$(x![y].P_1 \parallel x![w].P_2) \parallel x?[z].P_3 \longrightarrow (P_1 \parallel x![w].P_2) \parallel [y/z]P_3$$

- 違う環境の下でのプロセス同士が通信をすることがある．(有効範囲の移動)

ということがあげられる．これを解決するために，λ 計算のように局所的な計算のみを表現した簡約に加えて通信するプロセス同士が「近く」に来る，$\boldsymbol{\nu}$ を移動させるといった，プロセスの「配置換え」を許すようなプロセス間の等価関係 $\cong$:

$$(x![y].P_1 \parallel x![w].P_2) \parallel x?[z].P_3 \cong (x![y].P_1 \parallel x?[z].P_3) \parallel x![w].P_2$$

を導入する．

以下で，$P\uparrow_x$ は λ 計算でみたような，($\boldsymbol{\nu}$ の有効範囲にない) 変数参照 $\#^i x$ の添字 i を 1 増加させる操作，$[y/z]$ は (変数参照関係を保つ) 置き換えである．

2.2.1 定義: 構造的合同関係(*structural congruence*) $\Gamma \vdash P_1 \cong P_2$ [1]を以下の規則で定義する．

$$\frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash P \cong P}\ (\text{SC-Refl}) \qquad \frac{\Gamma \vdash P \in \mathbf{Pr} \quad \Gamma \vdash Q \in \mathbf{Pr} \quad \Gamma \vdash R \in \mathbf{Pr}}{\Gamma \vdash P \parallel (Q \parallel R) \cong (P \parallel Q) \parallel R}\ (\text{SC-ParAssoc})$$

$$\frac{\Gamma \vdash P \cong Q}{\Gamma \vdash Q \cong P}\ (\text{SC-Symm}) \qquad \frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash P + \mathbf{0} \cong P}\ (\text{SC-SumZero})$$

$$\frac{\Gamma \vdash P \cong Q \quad \Gamma \vdash Q \cong R}{\Gamma \vdash P \cong R}\ (\text{SC-Trans}) \qquad \frac{\Gamma \vdash P \in \mathbf{GPr} \quad \Gamma \vdash Q \in \mathbf{GPr}}{\Gamma \vdash P + Q \cong Q + P}\ (\text{SC-SumComt})$$

$$\frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash P \parallel \mathbf{0} \cong P}\ (\text{SC-ParZero}) \qquad \frac{\Gamma \vdash P \in \mathbf{GPr} \quad \Gamma \vdash Q \in \mathbf{GPr} \quad \Gamma \vdash R \in \mathbf{GPr}}{\Gamma \vdash P + (Q + R) \cong (P + Q) + R}\ (\text{SC-SumAssoc})$$

$$\frac{\Gamma \vdash P \in \mathbf{Pr} \quad \Gamma \vdash Q \in \mathbf{Pr}}{\Gamma \vdash P \parallel Q \cong Q \parallel P}\ (\text{SC-ParComt}) \qquad \frac{\Gamma \vdash P \in \mathbf{Pr}}{\Gamma \vdash *P \cong *P \parallel P}\ (\text{SC-Rep})$$

[1]多くの文献では $\equiv$ が使われている

$$\frac{\Gamma, x \vdash P \in \mathbf{Pr} \qquad \Gamma \vdash Q \in \mathbf{Pr}}{\Gamma \vdash ((\boldsymbol{\nu} x)P) \parallel Q \cong (\boldsymbol{\nu} x)(P \parallel Q{\uparrow_x})} \quad \text{(SC-NewExtr)}$$

$$\frac{\Gamma \vdash P \cong P' \qquad Q \cong Q'}{\Gamma \vdash P \parallel Q \cong P' \parallel Q'} \quad \text{(SC-Par)}$$

$$\frac{}{\Gamma \vdash (\boldsymbol{\nu} x)\mathbf{0} \cong \mathbf{0}} \quad \text{(SC-NewZero)}$$

$$\frac{\Gamma, x \vdash P \cong [x/y]Q}{\Gamma \vdash (\boldsymbol{\nu} x)P \cong (\boldsymbol{\nu} y)Q} \quad \text{(SC-New)}$$

$$\frac{\Gamma, x, y \vdash P \in \mathbf{Pr} \qquad (x \neq y)}{\Gamma \vdash (\boldsymbol{\nu} x)(\boldsymbol{\nu} y)P \cong (\boldsymbol{\nu} y)(\boldsymbol{\nu} x)P} \quad \text{(SC-NewSwap)}$$

2.2.2 定義: 簡約関係 $\Gamma \vdash P \longrightarrow Q$ を以下の規則で定義する.

$$\frac{\Gamma \vdash \tau.P + Q \in \mathbf{Pr}}{\Gamma \vdash \tau.P + Q \longrightarrow P} \quad \text{(R-Tau)}$$

$$\frac{\Gamma \vdash (\#^i x![\#^j z].P + R_1) \parallel (\#^i x?[y].Q + R_2) \in \mathbf{Pr}}{\Gamma \vdash (\#^i x![\#^j z].P + R_1) \parallel (\#^i x?[y].Q + R_2) \longrightarrow P \parallel [\#^j z/y]Q} \quad \text{(R-Comm)}$$

$$\frac{\Gamma \vdash P \longrightarrow Q \qquad \Gamma \vdash R \in \mathbf{Pr}}{\Gamma \vdash P \parallel R \longrightarrow Q \parallel R} \quad \text{(R-Par)}$$

$$\frac{\Gamma, x \vdash P \longrightarrow Q}{\Gamma \vdash (\boldsymbol{\nu} x)P \longrightarrow (\boldsymbol{\nu} x)Q} \quad \text{(R-New)}$$

$$\frac{\Gamma \vdash P \cong P' \qquad \Gamma \vdash P' \longrightarrow Q' \qquad \Gamma \vdash Q' \cong Q}{\Gamma \vdash P \longrightarrow Q} \quad \text{(R-SCong)}$$

2.3 非同期 π 計算

インターネットなど多くの現実的な分散システムでは，非同期 (asynchronous) 通信がより基本的な通信プリミティブとして用いられる．このような状況をモデル化するための体系が非同期 π 計算である．非同期 π 計算は，π 計算を，

- メッセージ送信後の動作
- 送信と他の動作の選択

を許さないよう制限することで簡単に得ることができる．具体的には，G-Out を P-Out で置き換えることになる．

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Ch} \qquad \Gamma \vdash \#^j y \in \mathbf{Ch}}{\Gamma \vdash \#^i x![\#^j y].\mathbf{0} \in \mathbf{Pr}} \quad \text{(P-Out)}$$

送信後の動作は必ず $\mathbf{0}$ とすることでひとつめの制限を，また，送信プロセスは $\mathbf{GPr}$ ではなく $\mathbf{Pr}$ とみなすことでふたつめの制限を表現している．

2.4 基本データ型の導入

純粋な λ 計算に整数，真偽値といった基本データ型を導入するのも比較的単純にできる．考えうる最も単純な拡張としては，送信される値として任意の式を許し，式の値の計算はプロセスの内部動作として考えられるので，

$$\frac{\Gamma \vdash a \Downarrow v \quad (a \text{ は値ではない})}{\Gamma \vdash \#^i x![a].P \longrightarrow \#^i x![v].P}$$

などの簡約を導入することが考えられる．

3 単純型付 π 計算

λ 計算と同様に，型システムを導入して，値の整合性 (送信に使われるチャネルが本当にチャネルであることなど) を検査することができる．ここでは，π 計算+算術式のための型システムとして自然数型とチャネル型のみを導入する．

3.1 型，型付構文，型判断，型付け規則

型:

$$\begin{array}{rcl} T & ::= & \mathtt{nat} \mid \updownarrow T \\ \Gamma & ::= & \bullet \mid \Gamma, x \in T \end{array}$$

$\updownarrow T$ は T を送受信するためのチャネル (の名前) を表す型である．

型付構文: $(\boldsymbol{\nu}(x \in T))P$

型判断: $\Gamma \vdash a \in T$ (a は変数参照を含む算術式とする) と $\Gamma \vdash P$ ok

型付け規則:

$$\Gamma \vdash \mathbf{0} \text{ ok} \qquad \text{(T-Nil)}$$

$$\frac{\Gamma \vdash P \text{ ok} \qquad \Gamma \vdash Q \text{ ok}}{\Gamma \vdash P \parallel Q \text{ ok}} \qquad \text{(T-Par)}$$

$$\frac{\Gamma, x \in \updownarrow T \vdash P \text{ ok}}{\Gamma \vdash (\boldsymbol{\nu}(x \in \updownarrow T))P \text{ ok}} \qquad \text{(T-New)}$$

$$\frac{\Gamma \vdash P \text{ ok}}{\Gamma \vdash *P \text{ ok}} \qquad \text{(T-Rep)}$$

$$\frac{\Gamma \vdash P_1 \text{ ok} \qquad \Gamma \vdash P_2 \text{ ok}}{\Gamma \vdash P_1 + P_2 \text{ ok}} \qquad \text{(T-Choice)}$$

$$\frac{\Gamma \vdash \#^i x \in \updownarrow T \qquad \Gamma \vdash P \text{ ok} \qquad \Gamma \vdash a \in T}{\Gamma \vdash \#^i x![a].P \text{ ok}} \qquad \text{(T-Out)}$$

$$\frac{\Gamma \vdash \#^i x \in \updownarrow T \qquad \Gamma, y \in T \vdash P \text{ ok}}{\Gamma \vdash \#^i x?[y].P \text{ ok}} \qquad \text{(T-In)}$$

$$\frac{\Gamma \vdash P \text{ ok}}{\Gamma \vdash \tau.P \text{ ok}} \qquad \text{(T-Tau)}$$

3.2 性質

3.2.1 定理 [Type Preservation]: $\Gamma \vdash P$ ok かつ $\Gamma \vdash P \longrightarrow P'$ ならば，$\Gamma \vdash P'$ ok である．

3.2.2 定理 [No Immediate Error]: $\Delta \vdash P$ ok ならば，

- $\Delta \vdash P \cong (\boldsymbol{\nu}(\vec{x} \in \vec{T}))(\mathtt{S(..S(0))}![a].Q + R_1) \parallel R_2$ なる，$\vec{x}, \vec{T}, a, Q, R_1, R_2$ は存在しない．
- $\Delta \vdash P \cong (\boldsymbol{\nu}(\vec{x} \in \vec{T}))(\mathtt{S(..S(0))}?[z].Q + R_1) \parallel R_2$ なる，$\vec{x}, \vec{T}, z, Q, R_1, R_2$ は存在しない．
- $\Delta \vdash P \cong (\boldsymbol{\nu}(\vec{x} \in \vec{T}))(\mathtt{\#}^i x![a].Q + R_1) \parallel R_2$ ならば $\Delta \vdash a \Downarrow \underbrace{\mathtt{S(..S}}_{n}\mathtt{(0)..)}$ なる自然数 n が存在する．

4 高階 π 計算

π 計算で表現される，チャネル名の受渡しによる通信トポロジーの変化は link mobility と呼ばれる．これに対して，プロセス自身の受渡しによる mobility (process mobility) を表現する体系が**高階 π 計算**(*higher-order π calculus*) である．

高階 π 計算では，$(x).P$ という形の (パラメータ抽象化された) プロセス—プロセスを λ 抽象したもの—と，抽象化プロセスの具体化—関数適用に対応する—を導入する．

型:

$$
\begin{array}{rcl}
T & ::= & \cdots \mid \updownarrow T \mid T \to \diamond \\
\Gamma & ::= & \bullet \mid \Gamma, x \in T
\end{array}
$$