The π-Calculus

Syntax, Semantics, and Undecidability

Concurrency Theory

Motivation: From Computability to Concurrency

- Classical undecidability: Turing machines, counter machines
- Question: can *pure interaction* cause undecidability?
- Answer: yes — via **mobility**

Central claim of this lecture:

> *The π-calculus is undecidable because it can dynamically create and reconfigure communication structure.*

Names and Intuition

Definition.

- Let $\mathcal{N}$ be a countably infinite set of *names*
- Names represent *channels*

Key principle:

- Names are values
- Names can be communicated
- Communication topology is dynamic

Syntax of the π-Calculus

Definition.

$$P ::= \mathbf{0} \mid \overline{a}\langle b\rangle.P \mid a(x).P \mid P \mid Q \mid (\nu a)P \mid !P \mid P + Q$$

All names $a, b, x \in \mathcal{N}$.

Meaning of Syntax (Complete)

$\mathbf{0}$	inactive process
$\overline{a}\langle b\rangle.P$	output b on channel a then behave as P
$a(x).P$	input on a, bind received name to x in P
$P \mid Q$	parallel composition
$(\nu a)P$	restrict name a (private scope)
$!P$	replication (infinitely many copies of P)
$P + Q$	nondeterministic choice

Free and Bound Names

Definition.

- $a(x).P$ binds x in P
- $(\nu a)P$ binds a in P

$$fn(P) = \text{names not bound in } P$$

Example:

$$(\nu a)(\overline{a}\langle b\rangle \mid a(x).P)$$

- a, x bound
- b free

Structural Congruence

Definition.
Structural congruence ($\equiv$) is the smallest congruence satisfying:

$$
\begin{aligned}
P \mid Q &\equiv Q \mid P \\
(P \mid Q) \mid R &\equiv P \mid (Q \mid R) \\
P \mid \mathbf{0} &\equiv P \\
(\nu a)(P \mid Q) &\equiv P \mid (\nu a)Q \quad (a \notin fn(P)) \\
(\nu a)(\nu b)P &\equiv (\nu b)(\nu a)P
\end{aligned}
$$

Purpose: enable communication and scope extrusion.

Operational Semantics: Labels

Definition.
The labelled transition system uses labels:

$$\alpha ::= \overline{a}\langle b\rangle \mid a(b) \mid \overline{a}(b) \mid \tau$$

- τ represents internal (unobservable) actions
- $\overline{a}(b)$ represents bound output (extrusion)

Operational Semantics: Core Rules

$$\overline{a}\langle b\rangle.P \xrightarrow{\overline{a}\langle b\rangle} P$$

$$a(x).P \xrightarrow{a(b)} P\{b/x\}$$

Substitution is capture-avoiding.

Communication and Scope Extrusion

$$\frac{P \xrightarrow{\overline{a}\langle b\rangle} P' \quad Q \xrightarrow{a(b)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$(\nu b)\overline{a}\langle b\rangle.P \xrightarrow{\overline{a}(b)} P$$

Scope extrusion: private names can escape their scope via communication.

Strong Bisimulation

Definition.
A relation $\mathcal{R}$ is a strong bisimulation if:

$$P \mathcal{R} Q \Rightarrow \forall \alpha,\ P \xrightarrow{\alpha} P' \Rightarrow \exists Q'.\ Q \xrightarrow{\alpha} Q' \wedge P' \mathcal{R} Q'$$

Strong bisimilarity: largest such relation.

Undecidability: Strategy Overview

- Use two-counter Minsky machines
- Encode machine states as processes
- Encode counters as name-based structures
- Reduce halting to termination

Two-Counter Machines

Definition.
A two-counter machine consists of:

- finite control states $q_0, \ldots, q_n$
- two counters $C_1, C_2 \in \mathbb{N}$

Instructions:

- $\mathrm{INC}(i)$
- $\mathrm{DEC}(i)$ / $\mathrm{ZERO?}(i)$
- HALT

Theorem. Halting is undecidable.

Encoding Counters in π

Counter value n encoded as:

$$C_i(n) \triangleq a_i \mid a_i \mid \cdots \mid a_i \quad (n \text{ copies})$$

Operations:

- INC: generate an additional a_i
- DEC: synchronize and consume one a_i
- ZERO?: branch on absence of a_i

Key idea: availability of communication encodes counter value.

Encoding Control States

Each machine state q_k encoded as a process:

$$State_k \triangleq \sum_j instr_{k,j}.State_j$$

- Instructions implemented by synchronization
- Choice models conditional branching

Complete Machine Encoding

$$P_M \triangleq (\nu \vec{a})(State_{init} \mid C_1 \mid C_2)$$

All names initially restricted.

Correctness of Encoding

Theorem.
Machine M halts iff P_M reaches a deadlocked configuration.
Proof (sketch).

- Each machine step simulated by finitely many π-steps
- Counter values preserved by synchronization discipline
- HALT state yields no enabled actions

Undecidability Results

Theorem. Termination of π-processes is undecidable.

Theorem. Strong bisimilarity of π-processes is undecidable.

Both follow by reduction from counter-machine halting.

Why CCS Remains Decidable

Feature	CCS	π
Name creation	$\times$	$\checkmark$
Name passing	$\times$	$\checkmark$
Dynamic topology	$\times$	$\checkmark$
Bisimulation	decidable	undecidable

Mobility is the source of undecidability.

Final Summary

- π-calculus models mobile concurrency
- Semantics via labelled transition systems
- Scope extrusion enables unbounded memory
- Counter machines encode directly
- Termination and bisimulation are undecidable