Lecture 2

Thursday, 9 February 2023 10:25

SPI calculus → successor of lambda calculus for reasoning about programs and processes

PI calculus → calculus for modelling concurrent processes

SPI calculus → extension of PI calculus with cryptographic primitives

simple syntax

operational semantics to model program execution

names n, m, l, k, c, d, e

variables x, y, z

variables are placeholders for names
names represent constants

messages M, N, L, K

basic process ; inactivity stop

$P \mid Q$ — P and Q executed in parallel

$!P$ — replication, i.e. infinite number of P in parallel

! binds stronger than |

if $M = N$ then P — continues with P if $M = N$

if binds stronger than |

first output M onto channel c, then return P — out c M ; P
$\bar{c}\langle M \rangle . P$

waits for a message to arrive on channel N, takes it out and binds it to variable x, then continue with process P — inp N x ; P
$N(x).P$

in inp N x ; P , the variable x is a binder whose scope is P
bound variables can be renamed
in this situation, occurrences of x in P are called bound
↑ i.e. in the same process only!

a variable occurrence is free if is neither a binder nor a bound variable

a process P is closed if no variable occurrence in P is free
a message M is closed if it does not contain variables

new n ; P
$(\nu n); P$ — generates a name n whose scope is P and continues with P
occurrence of n is a binder whose scope is P

bound names are private i.e. unknown to other/parallel processes
↓
but not free!

If M is a closed message, we define $\{M/x\} P$ as the process obtained by replacing all free occurrences of x in P by M

equivalent processes:

$P \mid Q = Q \mid P$

$(P \mid Q) \mid R = P \mid (Q \mid R)$

$P \mid stop = P$

$!P = P \mid !P$

these equivalence laws may be applied anywhere inside process expressions
this equivalence relation is called structural congruence

if n is not free in P, we have

$P \mid new\ n ; Q = new\ n ; (P \mid Q)$

this rule is called scope extrusion

→ Q is what remains of P after performing the step

$P \to Q$ — step relation on closed processes ; P can evolve into Q in one step

$P \to^{*} Q$ — transitive closure of step relation ; P can evolve into Q in zero or more steps
i.e. $P \equiv P_0 \to P_1 \to \ldots \to P_n \equiv Q$ for some $n \geq 0$
modulo structural congruence

[A1] if $P \equiv P_1$, $P_1 \to Q_1$ and $Q_1 \equiv Q$ then $P \to Q$

[A2] if $P \to P_1$, then $P \mid Q \to P_1 \mid Q$

step rule for new names: if $P \to Q$, then $new\ n ; P \to new\ n ; Q$

step rule for conditional: (if $M = M$ then P) $\to P$

step I/O-rule: $out\ c\ M ; P \mid inp\ c\ x ; Q \to P \mid \{M/x\} Q$

who consumes produced messages is non-deterministic