# A Process Calculus II

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC-Linz)
Johannes Kepler University, A-4040 Linz, Austria

Wolfgang.Schreiner@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/schreine

# Equational Laws

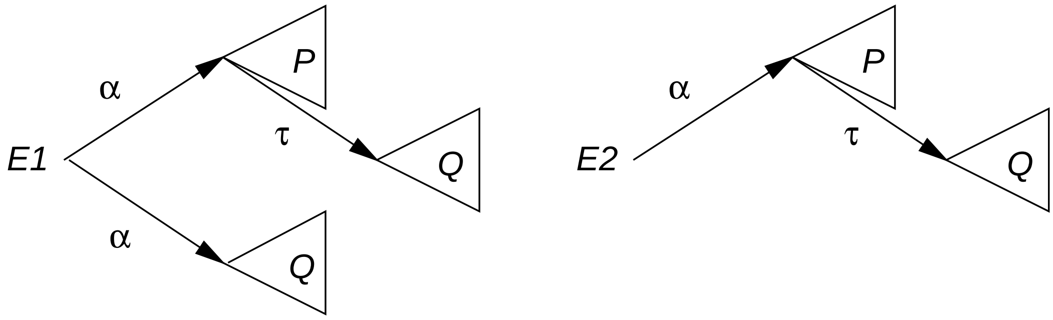
- Static laws
  - Static combinators: composition, restriction, labelling.
  - Action rules do not change graph structure.
  - Algebra of flow graphs.
- Dynamic laws
  - Dynamic combinators: prefix, summation, constants.
  - Action rules change graph structure.
  - Algebra of transition graphs.
- Expansion law
  - Relating static laws to dynamic laws.

# Static Laws

- Composition laws
  - $P|Q = Q|P$
  - $P + (Q|R) = (P|Q)|R$
  - $P|0 = P$
- Restriction laws
  - $P \backslash L = P$, if $L(P) \cap (L \cup \overline{L}) = \{ \}$.
  - $P \backslash K \backslash L = P \backslash (K \cup L)$
  - ...
- Relabelling laws
  - $P[\mathit{Id}] = P$
  - $P[f][f'] = P[f' \circ f]$
  - ...

# Dynamic Laws

- Monoid laws
  - $P + Q = Q + P$
  - $P + (Q + R) = (P + Q) + R$
  - $P + P = P$
  - $P + 0 = P$
- $\tau$ laws
  - $\alpha.\tau.P = \alpha.P$
  - $P + \tau.P = \tau.P$
  - $\alpha.(P + \tau.Q) + \alpha.Q = \alpha.(P + \tau.Q)$




- Transition Relation $P \stackrel{\alpha}{\Rightarrow} P'$
  - $P \stackrel{\alpha}{\Rightarrow} P' \equiv P(\stackrel{\tau}{\rightarrow})^* \stackrel{\alpha}{\rightarrow} (\stackrel{\tau}{\rightarrow})^* \; P'$

# Non-Laws

- $\tau.P = P$
  - A $= a.A + \tau.b.A$
  - A' $= a.A' + b.A'$
  - $A$ may switch to state in which only $b$ is possible.
  - $B$ *always* allows $a$ or $b$.
  - Action sequence $a, a$ may yield dedlock for right side.
- $\alpha.(P + Q) = \alpha.P + \alpha.Q$
  - $a.(b.P + c.Q) = a.b.P + a.c.Q$
  - $b.P$ is $a$-derivative of right side, not capable of $c$ action.
  - $a$-derivative of left side is capable of $c$ action!
  - Action sequence $a, c$ may yield deadlock for right side.

# The Expansion Law

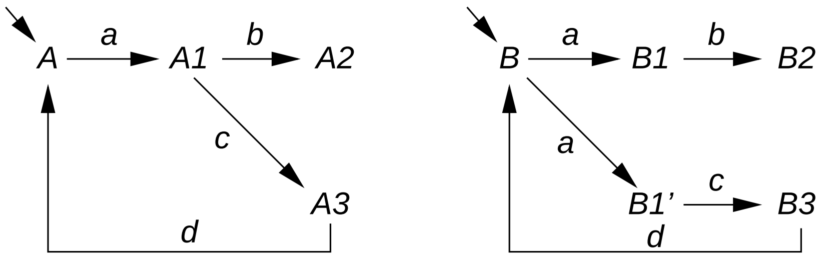
- The Expansion Law
  - Let $P \equiv (P_1[f_1]|\ldots|P_n[f_n])\backslash L$
  - $P = \Sigma\{f_1(\alpha).(P_1[f_1]|\ldots|P_i'[f_i]|\ldots|P_n[f_n])\backslash L$:
    $P_i \stackrel{\alpha}{\to} P_i'$, $f_i(\alpha)$ not in $L \cup \overline{\mathsf{L}}\}$
    $+ \Sigma\{\tau.(P_1[f_1]|\ldots|P_i'[f_i]|\ldots|P_j'[f_j]|\ldots|P_n[f_n])\backslash L$:
    $P_i \stackrel{l_1}{\to} P_i'$, $P_j \stackrel{l_2}{\to} P_j'$, $f_i(l_1) = \overline{f_i(l_2)}$, $i < j\}$
- Corollary
  - Let $P \equiv (P_1|\ldots|P_n)\backslash L$
  - $P = \Sigma\{\alpha.(P_1|\ldots|P_i'|\ldots|P_n)\backslash L$ :
    $P_i \stackrel{\alpha}{\to} P_i'$, $\alpha$not in $L \cup L'\}$
    $+ \Sigma\{\tau.(P_1|\ldots|P_i'|\ldots|P_j'|\ldots P_n)\backslash L$ :
    $P_i \stackrel{\mathsf{l}}{\to} P_i'$, $P_j \stackrel{\overline{\mathsf{l}}}{\to} P_j'$, $i < j\}$
- Example
  - $P_1 = a.P_1' + b.P_1''$
  - $P_2 = \overline{a}.P_2' + c.P_2''$
  - $(P_1|P_2)\backslash a = b.(P_1''|P_2)\backslash a + c.(P_1|P_2'')\backslash a + \tau.(P_1'|P_2')\backslash a$

# Equivalence of Agents

- Equivalence
  - Two agents $P$ and $Q$ are different if distinction can be detected by external agent interacting with them
- *Strong* equivalence:
  - $\tau$ is treated like any other (observable) action.
- *Observation* equivalence:
  - $\tau$ cannot be observed by external agent.
  - *Congruence relation* i.e. preserved by all algebraic contexts.

# Experimenting upon Agents

- Example agents $A$ and $B$
  - $A = a.(b.0 + c.d.A)$
  - $B = a.b.0 + a.c.d.B$

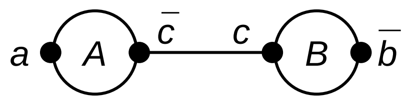


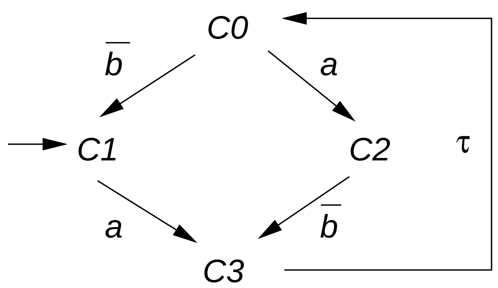

- "Language understood" by $A$ and $B$
  - $(a.c.d)^*.a.b.0$
  - $A$ and $B$ seem equivalent.
- Ports $a$, $b$, $c$, $d$.
  - Initially only $a$ is "unlocked".
  - Observer "presses button" $a$.
  - In $A$, $b$ and $c$ are "unlocked".
  - In $B$, sometimes $b$, sometimes $c$ is "unlocked".
  - $A$ and $B$ can be experimentally distinguished!

# Strong Bisimulation

- *Strong bisimulation*
  - Binary relation $S$ over agents such that $(P, Q) \in S$ implies
  - If $P \stackrel{\alpha}{\rightarrow} P'$, then $Q \stackrel{\alpha}{\rightarrow} Q'$ with $(P', Q') \in S$ and vice versa.
  - For every action $\alpha$, every $\alpha$-derivative of $P$ is equivalent to some $\alpha$-derivative of $Q$.

- Example




  - Claim: $(A|B)\backslash c = C_1$
  - True if $S$ is a strong bisimulation:
    $S = \{ ((A|B)\backslash c,\ C_1),\ ((A'|B)\backslash c,\ C_3),$
    $((A|B')\backslash c,\ C_0),\ ((A'|B')\backslash c,\ C_2) \}$
  - Check derivatives of each of the eight agents.

# Strong Equivalence

- *Strong equivalence* $P{\sim}Q$
  - $P{\sim}Q$, if $(P, Q) \in S$ for some strong bisimulation $S$.
  - $\sim\; = \cup\{S\text{: } S \text{ is a strong bisimulation}\}$.
- Corollaries:
  - $\sim$ is the largest strong bisimulation.
  - $\sim$ is an equivalence relation.
- Proposition:
  - $P{\sim}Q$ iff, for all $\alpha$,
  - If $P \stackrel{\alpha}{\to} P'$, then $Q \stackrel{\alpha}{\to} Q'$ with $(P', Q') \in S$ and vice versa.

# Properties of Strong Equivalence

- Monoid laws:
  - $P + Q \sim Q + P$
  - $P + (Q + R) \sim (P + Q) + R$
  - $P + P \sim P$
  - $P + 0 \sim P$
- Static laws:
  - $P|Q \sim Q|P$
  - $P + (Q|R) \sim (P|Q)|R$
  - $P|0 \sim P$
  - $\ldots$
- The Expansion law:
  - $\ldots$ (= replaced by $\sim$)

# Strong Congruence

- Strong congruence
  - Strong equivalence is substitutive under all combinators and under recursive definitions
- Let $P_1 \sim P_2$
  - $\alpha.P_1 \sim \alpha.P_2$
  - $P_1 + Q \sim P_2 + Q$
  - $P_1|Q \sim P_2|Q$
  - $P_1\backslash L \sim P_2\backslash L$
  - $P_1[f] \sim P_2[f]$

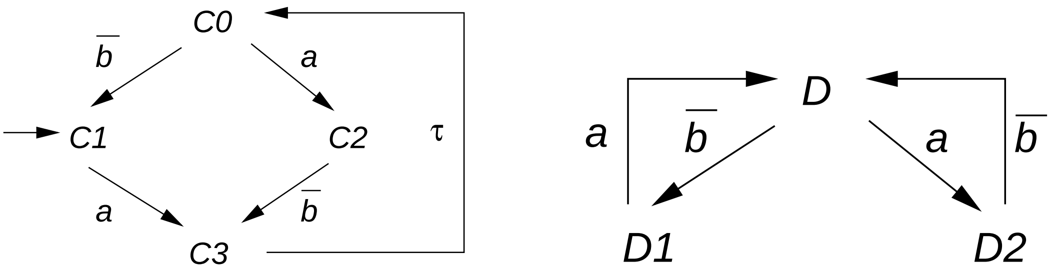
# Bisimulation and Observation Equivalence

- (Weak) bisimulation and (observation) equivalence:
  - $\tau$ action may be matched by zero or more $\tau$ actions.
- Auxiliary definitions:
  - $\hat{t}$ is the action sequence gained by deleting all occurences of $\tau$ from $t$.
  - $E \stackrel{t}{\rightarrow} E'$, if $t = \alpha_1 \ldots \alpha_n$ and $E \stackrel{\alpha_1}{\rightarrow} \ldots \stackrel{\alpha_n}{\rightarrow} E'$.
  - $E \stackrel{t}{\Rightarrow} E'$ if $t = \alpha_1 \ldots \alpha_n$ and $E (\stackrel{\tau}{\rightarrow})^* \stackrel{\alpha_1}{\rightarrow} (\stackrel{\tau}{\rightarrow})^* \ldots (\stackrel{\tau}{\rightarrow})^* \stackrel{\alpha_n}{\rightarrow} (\stackrel{\tau}{\rightarrow})^* E'$.
  - $E'$ is a $t$-descendant of $E$ iff $E \stackrel{\hat{t}}{\Rightarrow} E'$.
- Relationship
  - $P \stackrel{t}{\rightarrow} P'$ implies $P \stackrel{t}{\Rightarrow} P'$ implies $P \stackrel{\hat{t}}{\rightarrow} P'$

# Weak Bisimulation and Observation Equivalence

- *(Weak) bisimulation*
  - Binary relation $S$ such that $(P, Q) \in S$ implies
  - if $P \stackrel{\alpha}{\to} P'$, then $Q \stackrel{\widehat{\alpha}}{\Rightarrow} Q'$ with $(P', Q') \in S$ (and vice versa).
- *Observation equivalence* $P \approx Q$
  - $P \approx Q$ if $(P, Q) \in S$ for some weak bisimulation $S$.
  - $\approx\ = \cup\{S : S$ is a weak bisimulation$\}$

# Examples




- Agents $C_0$ and $D$
  - Bisimulation $S =$ $\{(C_0, D), (C_1, D_1), (C_2, D_2), (C_3, D)\}$
  - No *strong* bisimulation containing $(C_3, D)$ since $C_3 \stackrel{\tau}{\rightarrow} C_0$ but there is no $D \stackrel{\tau}{\rightarrow} D'$.
- Agents $A$ and $B$
  - $A_0 = a.A_0 + b.A_1 + \tau.A_1$
    $A_1 = a.A_1 + \tau.A_2$
    $A_2 = b.A_0$
  - $B_1 = a.B_1 + \tau.B_2$
    $B_2 = b.B_1$
  - Bisimulation $S = \{ (A_0, B_1), (A_1, B_1), (A_2, B_2) \}$ (note that $B_1 \stackrel{b}{\Rightarrow} B_1$!)

# Properties of Bisimulation

- Propositions:
  - $\approx$ is the largest bisimulation.
  - $\approx$ is an equivalence relation.
  - $P \approx \tau.P$
- $\approx$ is *not* yet equality:
  - $\approx$ not preserved by summation.
  - $a.0 + b.0 \approx a.0+\tau.b.0$ does *not* hold!
  - Proof: if $(P,Q)$ were in a bisimulation $S$, then, since $Q \stackrel{\tau}{\to} b.0$, we need $(P', b.0)$ in $S$ with $P \stackrel{\epsilon}{\Rightarrow} Q'$. But the only $P'$ is $P$ itself but $(P, b.0)$ can be not in $S$, since $P \stackrel{a}{\to}$ 0, while $b.0$ has no $a$-descendant.
- Relations:
  - $P{\sim}Q$ implies $P{=}Q$ implies $P{\approx}Q$

*Equality not yet fully captured.*

# Observation Congruence

- *Stability*:
  - $P$ is stable if $P$ has no $\tau$-derivative.
- Derivation of equality:
  - If $P \approx Q$ and both are stable, then $P = Q$.
  - If $P \approx Q$ then $\alpha.P = \alpha.Q$
- $P = Q$ (*observation congruence*)
  - If $P \stackrel{\alpha}{\rightarrow} P'$, then $Q \stackrel{\alpha}{\Rightarrow} Q'$ with $P' \approx Q'$ (and vice versa).
  - Preserved under all process operators.

*Observation congruence is the equality of the process algebra.*

# Summary

- Algebraic approach to semantics of parallel programs.
  - Processes are algebraic terms.
  - Calculus for term manipulation preserving equality.
  - Main interest: how do processes interact with each other?
- Central notions:
  - Strong bisimilarity: equivalence even for internal actions.
  - Observation equivalence: equivalence only for observable actions.
  - Observation congruence: equivalence preserved under all substitutions.

*Modeling of systems that react with their environment.*