The essence of
PARALLEL ALGOL

Stephen Brookes

Department of Computer Science
Carnegie Mellon University

LICS '96

ESSENTIALS

- PARALLEL ALGOL =

  shared-variable parallel programs

  + call-by-name λ-calculus

- simply typed

  $$\theta ::= \mathbf{exp}[\tau] \mid \mathbf{var}[\tau] \mid \mathbf{comm}$$
  $$\mid (\theta \to \theta') \mid \theta \times \theta'$$ *phrase types*

  $$\tau ::= \mathbf{int} \mid \mathbf{bool}$$ *data types*

- recursion and conditional at each type

cf. Reynolds: *The essence of* ALGOL

RATIONALE

- Can write parallel programs that cooperate by reading and writing shared memory

- Procedures can encapsulate parallel idioms (e.g. mutual exclusion, readers–writers)

- Local variable declarations can be used to limit the scope of interference

INTUITION

Procedures and parallelism are *orthogonal*:

- should combine smoothly
- semantics should be “modular”
- should obtain a conservative extension

MUTUAL EXCLUSION

procedure $mutex(n_1, c_1, n_2, c_2)$;
boolean s;
begin
    s:=**true**;
    while true do
        $(n_1;$ **await** s **then** s:=**false**;
        $c_1;$ s:=**true**$)$
    $\|$ **while true do**
        $(n_2;$ **await** s **then** s:=**false**;
        $c_2;$ s:=**true**$)$
end

- Encapsulates common use of a *semaphore*
- Correctness relies on *locality* of s
- Independent of n_i and c_i

OUTLINE of SEMANTICS

- Traditional "global state" models fail to validate natural equivalences, e.g.

$$\mathbf{new}[\tau]\ \iota\ \mathbf{in}\ P = P$$

  when ι does not occur free in P.

- We adapt "possible worlds" model of sequential ALGOL to the parallel setting...
- ...and simultaneously extend our "transition trace" semantics (LICS'93) to include procedures and recursion.

- We adapt a "relationally parametric" model of sequential ALGOL to the parallel setting...
- ...and introduce a form of parametric reasoning for shared-variable programs.

cf. Reynolds, Oles

cf. O'Hearn, Tennent

CATEGORY of WORLDS

- Objects are countable sets (of "allowed states")
- Morphisms are "expansions":

$$h = (f, Q) : W \to X$$

  where

  - f is a function from X to W
  - Q is an equivalence relation on X
  - f puts each Q-class in bijection with W

INTUITION

- X is a set of "large" states extending the "small" states of W
- f extracts the "small" part of a state
- Q identifies states with the same extra parts

cf. Frank Oles' *Ph.D. thesis*

EXPANSIONS

- For each pair of objects W and V there is a canonical *expansion* morphism

  $$- \times V : W \to W \times V$$

  given by

  $$- \times V = (\text{fst} : W \times V \to W,\ Q)$$

  where

  $$((w_0, v_0), (w_1, v_1)) \in Q \iff v_0 = v_1$$

- Every morphism is such an expansion composed with an isomorphism.

INTUITION

An expansion $- \times V_\tau$ models the introduction of a local variable of datatype τ.

SEMANTICS

- Types denote functors from worlds to domains:

$$[\![\theta]\!] : \mathbf{W} \to \mathbf{D}$$

- Phrases denote natural transformations:

$$[\![P]\!] : [\![\pi]\!] \stackrel{\cdot}{\to} [\![\theta]\!]$$

i.e. when $h : W \to X$,

$$\begin{array}{ccc} [\![\pi]\!]W & \xrightarrow{[\![P]\!]W} & [\![\theta]\!]W \\ {\scriptstyle [\![\pi]\!]h}\downarrow & & \downarrow{\scriptstyle [\![\theta]\!]h} \\ [\![\pi]\!]X & \xrightarrow[{[\![P]\!]X}]{} & [\![\theta]\!]X \end{array}$$

commutes.

When h is an expansion naturality enforces locality.

CARTESIAN CLOSURE

- The functor category $\mathbf{D}^{\mathbf{W}}$ is cartesian closed.
- Can use ccc structure to interpret arrow types.

Procedures of type $\theta \to \theta'$ denote, at world W, natural families of functions $p(-)$:

- When $h : W \to X$ and $h' : X \to Y$,

$$\begin{array}{ccc}
[\![\theta]\!]X & \xrightarrow{\quad p(h) \quad} & [\![\theta']\!]X \\
{\scriptstyle [\![\theta]\!]h'} \downarrow & & \downarrow {\scriptstyle [\![\theta']\!]h'} \\
[\![\theta]\!]Y & \xrightarrow[\quad p(h;h') \quad]{} & [\![\theta']\!]Y
\end{array}$$

commutes.

INTUITION

Procedures can be called at expanded worlds, but naturality enforces locality constraints.

COMMANDS

- Commands denote sets of *traces*:

$$[\![\textbf{comm}]\!]W = \wp^{\dagger}((W \times W)^{\infty})$$

- Trace sets are *closed*, e.g.
  - $\alpha\beta \in c \,\&\, w \in W \;\Rightarrow\; \alpha(w, w)\beta \in c$
  - $\alpha(w, w')(w', w'')\beta \in c \;\Rightarrow\; \alpha(w, w'')\beta \in c$

- When $h : W \to X$, $[\![\textbf{comm}]\!]h$ converts a trace set over W to a trace set over X:

$$[\![\textbf{comm}]\!](f, Q)c = \{\beta \mid \text{map}(f \times f)\beta \in c \,\&\, \text{map}(Q)\beta\}$$

INTUITION

- A trace $(w_0, w_0')(w_1, w_1')\ldots(w_n, w_n')$ represents a fair interactive computation.
- Each step (w_i, w_i') represents a finite sequence of atomic actions.
- $[\![\textbf{comm}]\!]hc$ behaves like c on the *W*-component of state and has no effect elsewhere.

EXPRESSIONS

Expressions denote trace sets:

$$[\![\mathbf{exp}[\tau]]\!]W = \wp^{\dagger}(W^{+} \times V_{\tau} \ \cup \ W^{\omega})$$

$$\begin{aligned}[\![\mathbf{exp}[\tau]]\!](f, Q)e = &\{(\rho', v) \mid (\mathrm{map} f \rho', v) \in e\} \\ &\cup \{\rho' \mid \mathrm{map} f \rho' \in e \cap W^{\omega}\}\end{aligned}$$

VARIABLES

"Object-oriented" interpretation *à la* Reynolds:

variable = acceptor + expression

$$[\![\mathbf{var}[\tau]]\!]W = (V_{\tau} \to [\![\mathbf{comm}]\!]W) \times [\![\mathbf{exp}[\tau]]\!]W$$

RECURSION

Requires a careful use of *greatest fixed points*:

- Embed $\llbracket\theta\rrbracket W$ in a complete lattice $[\theta]W$ (like $\llbracket\theta\rrbracket W$ but without closure and naturality)

- Generalize semantic definitions to $[P]W$.

- Introduce natural transformations

$$\text{stut}_\theta : [\theta] \stackrel{\cdot}{\rightarrow} [\theta] \qquad \text{clos}_\theta : [\theta] \stackrel{\cdot}{\rightarrow} \llbracket\theta\rrbracket$$

- Can then define $\llbracket\textbf{rec}\ \iota.P\rrbracket Wu$ to be

$$\text{clos}_\theta W(\nu x.\text{stut}_\theta W([P]W(u \mid \iota : x)))$$

EXAMPLE

- Divergence = infinite stuttering:

$$\begin{aligned}\llbracket\textbf{rec}\ \iota.\iota\rrbracket Wu &= (\nu c.\{(w,w)\alpha \mid \alpha \in c\})^\dagger \\ &= \{(w,w) \mid w \in W\}^\omega\end{aligned}$$

LAWS

- This semantics validates:

$$\textbf{new}[\tau]\ \iota\ \textbf{in}\ P' = P'$$
$$\textbf{new}[\tau]\ \iota\ \textbf{in}\ (P \| P') = (\textbf{new}[\tau]\ \iota\ \textbf{in}\ P) \| P'$$
$$\textbf{new}[\tau]\ \iota\ \textbf{in}\ (P; P') = (\textbf{new}[\tau]\ \iota\ \textbf{in}\ P); P'$$

  when ι does not occur free in P'.

- Also (still) validates:

$$(\lambda\iota : \theta.P)(Q) = P[Q/\iota]$$
$$\textbf{rec}\ \iota.P = P[\textbf{rec}\ \iota.P/\iota]$$

- Orthogonal combination of laws of shared-variable programming with laws of λ-calculus.

PROBLEM

Semantics fails to validate

$$\textbf{new}[\textbf{int}]\ \iota\ \textbf{in}\ (\iota{:=}0;\ P(\iota{:=}\iota + 1)) = P(\textbf{skip}),$$

where P is a free identifier of type **comm** $\rightarrow$ **comm**.

REASON

- Equivalence proof relies on relational reasoning.
- Naturality does not enforce enough constraints on procedure meanings.

SOLUTION

- Same problem arose in sequential setting.
- Develop a relationally parametric semantics…

cf. O'Hearn and Tennent

PARAMETRIC MODEL

- Category of relations $R : W_0 \leftrightarrow W_1$
- A morphism from R to S is a pair (h_0, h_1) of morphisms in **W** such that

$$\begin{array}{ccc} W_0 & \xrightarrow{h_0} & X_0 \\ R \updownarrow & & \updownarrow S \\ W_1 & \xrightarrow[h_1]{} & X_1 \end{array}$$

- Types denote *parametric* functors, e.g.
  - if $R : W_0 \leftrightarrow W_1$, $[\![\theta]\!]R : [\![\theta]\!]W_0 \leftrightarrow [\![\theta]\!]W_1$
  - $(d_0, d_1) \in [\![\theta]\!]R \Rightarrow ([\![\theta]\!]h_0 d_0, [\![\theta]\!]h_1 d_1) \in [\![\theta]\!]S$

- Phrases denote *parametric* natural transformations:

  $(u_0, u_1) \in [\![\pi]\!]R \;\Rightarrow\; ([\![P]\!]W_0 u_0, [\![P]\!]W_1 u_1) \in [\![\theta]\!]R$

- The *parametric functor* category is cartesian closed.

COMMANDS

When $R : W_0 \leftrightarrow W_1$ define:

$$(c_0, c_1) \in [\![\mathbf{comm}]\!]R \iff$$

$$\begin{array}{l} \forall(\rho_0, \rho_1) \in \text{map}(R). \\ \quad [\forall\alpha_0 \in c_0.\ \text{map fst}\ \alpha_0 = \rho_0 \ \Rightarrow \\ \quad\ \exists\alpha_1 \in c_1.\ \text{map fst}\ \alpha_1 = \rho_1 \ \& \\ \qquad\qquad (\text{map snd}\ \alpha_0,\ \text{map snd}\ \alpha_1) \in \text{map}(R)] \\ \ \& \\ \quad [\forall\alpha_1 \in c_1.\ \text{map fst}\ \alpha_1 = \rho_1 \ \Rightarrow \\ \quad\ \exists\alpha_0 \in c_0.\ \text{map fst}\ \alpha_0 = \rho_0 \ \& \\ \qquad\qquad (\text{map snd}\ \alpha_0, \text{map snd}\ \alpha_1) \in \text{map}(R)]. \end{array}$$

This is parametric!

INTUITION

When related commands are started and interrupted in related states their responses are related.

LAWS

- As before,

$$\begin{aligned}&\textbf{new}[\tau]\ \iota\ \textbf{in}\ P' = P'\\&\textbf{new}[\tau]\ \iota\ \textbf{in}\ (P\|P') = (\textbf{new}[\tau]\ \iota\ \textbf{in}\ P)\|P'\\&\textbf{new}[\tau]\ \iota\ \textbf{in}\ (P; P') = (\textbf{new}[\tau]\ \iota\ \textbf{in}\ P); P'\end{aligned}$$

when ι does not occur free in P'.

- As before,

$$\begin{aligned}&(\lambda\iota : \theta.P)Q = [Q/\iota]P\\&\textbf{rec}\ \iota.P = [\textbf{rec}\ \iota.P/\iota]P\end{aligned}$$

- In addition,

$$\begin{aligned}&\textbf{new}[\textbf{int}]\ \iota\ \textbf{in}\ (\iota{:=}1; P(\iota)) \ = \ P(1)\\&\textbf{new}[\textbf{int}]\ \iota\ \textbf{in}\ (\iota{:=}0;\ P(\iota{:=}\iota+1)) \ = \ P(\textbf{skip}),\end{aligned}$$

relying crucially on parametricity.

EXAMPLE

$$\begin{array}{l}\textbf{new}[\textbf{int}]\ x\ \textbf{in}\\ \quad (x{:=}0;\ P(x{:=}x+1; x{:=}x+1);\\ \quad \textbf{if}\ even(x)\ \textbf{then diverge else skip})\end{array}$$

and

$$\begin{array}{l}\textbf{new}[\textbf{int}]\ x\ \textbf{in}\\ \quad (x{:=}0;\ P(x{:=}x+2);\\ \quad \textbf{if}\ even(x)\ \textbf{then diverge else skip})\end{array}$$

are equivalent in sequential ALGOL
but not equivalent in PARALLEL ALGOL.

The relation

$$(w,(w',z)) \in R \iff w = w' \,\&\, even(z)$$

works for sequential model but not for parallel.

CONCLUSIONS

- Can combine parallelism and procedures smoothly:
  - faithful to the essence of ALGOL
  - allows formalization of parallel idioms
  - retains laws of component languages
- Semantics by "modular" combination:
  - traces + possible worlds
  - traces + relational parametricity
- Advantages:
  - full abstraction at ground types
  - supports common reasoning principles:
    - representation independence
    - global invariants
    - assumption–commitment
- Limitations:
  - does not build in irreversibility of state change

SEMANTICS of skip

Finite stuttering:

$$\begin{aligned} [\![\textbf{skip}]\!]Wu &= \{(w,w) \mid w \in W\}^\dagger \\ &= \{(w,w) \mid w \in W\}^+ \end{aligned}$$

ASSIGNMENT

Non-atomic; source expression evaluated first:

$$\begin{aligned} &[\![I{:=}E]\!]Wu = \\ &\qquad \{(\mathrm{map}\Delta_W \rho)\beta \mid (\rho, v) \in [\![E]\!]Wu \\ &\qquad\qquad \;\&\; \beta \in \mathrm{fst}([\![I]\!]Wu)v\}^\dagger \\ &\quad \cup \{\mathrm{map}\Delta_W \rho \mid \rho \in [\![E]\!]Wu \cap W^\omega\}^\dagger. \end{aligned}$$

PARALLEL COMPOSITION

$$[\![P_1 \| P_2]\!]Wu = \{\alpha \mid \exists \alpha_1 \in [\![P_1]\!]Wu,\ \alpha_2 \in [\![P_2]\!]Wu. \\ (\alpha_1, \alpha_2, \alpha) \in \mathit{fairmerge}_{W \times W}\}^{\dagger}$$

where

$$\mathit{fairmerge}_A = \mathit{both}_A^* \cdot \mathit{one}_A \cup \mathit{both}_A^\omega$$
$$\mathit{both}_A = \{(\alpha, \beta, \alpha\beta), (\alpha, \beta, \beta\alpha) \mid \alpha, \beta \in A^+\}$$
$$\mathit{one}_A = \{(\alpha, \epsilon, \alpha), (\epsilon, \alpha, \alpha) \mid \alpha \in A^\infty\}$$

LOCAL VARIABLES

$$[\![\mathbf{new}[\tau]\ \iota\ \mathbf{in}\ P]\!]Wu = \{\mathrm{map}(\mathrm{fst} \times \mathrm{fst})\alpha \mid \\ \mathrm{map}(\mathrm{snd} \times \mathrm{snd})\alpha \text{ interference-free } \& \\ \alpha \in [\![P]\!](W \times V_\tau)([\![\pi]\!](- \times V_\tau)u \mid \iota : (a, e))\}$$

- No external changes to local variable
- $(a, e) \in [\![\mathbf{var}[\tau]]\!](W \times V_\tau)$ is a "fresh variable" corresponding to the V_τ component of the state

AWAIT

$$[\![\textbf{await } B \textbf{ then } P]\!]Wu =$$
$$\{(w,w') \in [\![P]\!]Wu \mid (w,\mathtt{tt}) \in [\![B]\!]Wu\}^\dagger$$
$$\cup \{(w,w) \mid (w,\mathtt{ff}) \in [\![B]\!]Wu\}^\omega$$
$$\cup \{\mathrm{map}\Delta_W \rho \mid \rho \in [\![B]\!]Wu \cap W^\omega\}^\dagger.$$

- P is atomic, enabled only when B true.
- Busy wait when B false.

λ-CALCULUS

$$[\![\iota]\!]Wu = u\iota$$

$$[\![\lambda\iota : \theta.P]\!]Wuha = [\![P]\!]W'([\![\pi]\!]hu \mid \iota : a)$$

$$[\![P(Q)]\!]Wu = [\![P]\!]Wu(\mathrm{id}_W)([\![Q]\!]Wu),$$

- This is the standard interpretation, based on the ccc structure of the functor category.