Axiomatic Semantics

Basics

An execution consists of shared memory access events and relations among the events.

Axioms provides certain consistency constraints on the events and relations.

A *consistent* execution satisfies all the constraints.

Program semantics is represented by a set of consistent executions.

Program behavior is represented by the set of outcomes of the consistent executions.

An execution is represented as a graph where nodes denote events and edges denote relations among the events.

Event

- An event $e = \langle uid, i, \ell \rangle$ is a tuple where
  - uid $\in \mathbb{N}$ is an unique identifier,
  - $i \in \mathsf{Tid} \cup \{0\}$ is thread identifier, and
  - ℓ is a label

- Label $\ell = \langle \mathsf{op}, \mathsf{loc}, \mathsf{rVal}, \mathsf{wVal} \rangle$ is a tuple where op, loc, rVal, wVal denote operation, location, read value, and written value respectively.

- e.uid, e.tid, e.op, e.loc, e.rVal, and e.wVal denote the identifier, thread id, operation, location, read value, and written value of event e.

- Representation of an event:
  - Read event $e = \mathsf{R}(X, v)$ as $e.\mathsf{wVal} = \perp$
  - Write event $e = \mathsf{W}(X, v)$ as $e.\mathsf{rVal} = \perp$
  - Update event $e = \mathsf{U}(X, v_r, v_w)$
  - Fence event $e = \mathsf{F}$ as $e.\mathsf{loc} = \perp$, $e.\mathsf{rVal} = \perp$, and $e.\mathsf{wVal} = \perp$

Relations

Each relation relates a pair of events.

Two types of relations: primitive and derived

Primitive relations are components of execution. For example

- program-order (po)
- read-from rf

Derived relations are derived from other primitive or derived relations. For example

- happens-before (hb)

Execution

An execution $\langle E, \mathrm{po}, \mathrm{rf} \rangle$ is a tuple where:

- E is a finite set of events
- po is a partial order on E. Denotes the syntactic order among events.
- rf is a binary relation on E such that for all $(w, r) \in \mathrm{rf}$
  - $w.\mathrm{op} = \mathrm{W}$ or $w.\mathrm{op} = \mathrm{U}$
  - $r.\mathrm{op} = \mathrm{R}$ or $r.\mathrm{op} = \mathrm{U}$
  - $w.\mathrm{loc} = r.\mathrm{loc}$
  - $w.\mathrm{wVal} = r.\mathrm{rVal}$

Execution Graph

Ler $G = \langle E, \mathsf{po}, \mathsf{rf} \rangle$ be an execution graph.

- $G.\mathsf{E} = \mathsf{E}$, set of events
- $G.\mathsf{po} = \mathsf{po}$, set of po edges
- $G.\mathsf{rf} = \mathsf{rf}$, set of rf edges

Moreover,

- $G.\mathsf{R} = \{a \mid a.\mathsf{op} = \mathsf{R}\}$
- $G.\mathsf{W} = \{a \mid a.\mathsf{op} = \mathsf{W}\}$
- $G.\mathsf{U} = \{a \mid a.\mathsf{op} = \mathsf{U}\}$
- $G.\mathsf{F} = \{a \mid a.\mathsf{op} = \mathsf{F}\}$
- $G.\mathsf{WU} = G.\mathsf{W} \cup G.\mathsf{U}$
- $G.\mathsf{RU} = G.\mathsf{R} \cup G.\mathsf{U}$
- $G.\mathsf{E}^i = \{a \mid a \in G.\mathsf{E} \wedge a.\mathsf{Tid} = i\}$
- $G.\mathsf{E}_x = \{a \mid a \in G.\mathsf{E} \wedge a.\mathsf{loc} = x\}$
- $\ldots$

Thread Graph

Execution of a thread i results in respective sequential graph G^i.

(Silent)

$$\frac{c, s \xrightarrow{\epsilon} c', s'}{c, s, G \Rightarrow c', s', G}$$

(Non-silent)

$$\frac{\begin{array}{c} c, s \xrightarrow{\ell} c', s' \\ a = \langle n, i, \ell \rangle \\ \forall e \in G.\mathrm{E}\ e.\mathrm{uid} \neq n \end{array}}{c, s, G \Rightarrow c', s', G'}$$

where

$G'.\mathrm{E} = G.\mathrm{E} \cup \{a\}$

$G'.\mathrm{po} = G.\mathrm{po} \cup (G.\mathrm{E}^i \times \{a\})$

$G'.\mathrm{rf} = G.\mathrm{rf}$

Program and Execution

G is an execution graph of a program P if G^i is an execution of $P(i)$ for every thread $i \in \mathsf{Tid}$.

Complete Execution Graph

An execution graph is complete if every read or event reads from a write or update event.

$\forall r \in G.\mathsf{RU}.\ \exists w \in G.\mathsf{WU}.\ (w, r) \in G.\mathsf{rf}$

Alternatively, $\mathsf{codom}(G.\mathsf{rf}) = G.\mathsf{RU}$

Consistent Execution

Let MM be a set of consistency constraints.
$\mathrm{Cons}(G, \mathrm{MM}) =$ true if a complete execution graph G satisfies all MM constraints.

Semantics of a Program

Semantics of a program P in consistency model MM is is denoted by the set of of its MM consistent executions.

$[\![P, \mathrm{MM}]\!] = \{G \mid \mathrm{Cons}(G, \mathrm{MM})\}$