Basic declarative semantics of concurrency

Homework assignment

November 16, 2023

Abstract

This assignment is dedicated to the declarative semantics of weak memory models. Its goal is to reinforce the concepts of executions graphs, various relations on these graphs, and the consistency predicates.

Introduction

Recall the definition of the *execution graphs*.

Definition 1. An *execution graph* G is a tuple $\langle \mathtt{E}, \mathtt{po}, \mathtt{rf}, \mathtt{co} \rangle$, which components are defined as follows.

- $\mathtt{E} \subseteq \mathbb{N} \times \mathsf{Tid} \times \mathsf{Lab}$ is a set of events.
  Event $e = \langle n, t, \ell \rangle$ is a triple, where:
  - n is an event identifier;
  - t is a thread identifier;
  - ℓ is a label.
- $\mathtt{po} \subseteq \mathtt{E} \times \mathtt{E}$ is the *program order* relation. It is a strict partial order that totally orders all the events within a single thread.
- $\mathtt{rf} \subseteq [\mathtt{W}]; =_{\mathtt{loc}} \cap =_{\mathtt{val}}; [\mathtt{R}]$ is the *reads-from* relation. This relation connects each write event with the read events that read value from it.

- $\texttt{co} \subseteq [\texttt{W}]; =_{\texttt{loc}}; [\texttt{W}]$ is the *coherence order* relation. This is a strict partial order that totally orders all write operations to the same location.

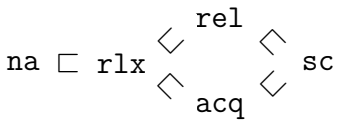
The definition of labels is the same as was given in previous assignment, except we added the access mode parameter. A label $\ell \in \mathsf{Lab}$ can be one of the following:

- $\texttt{R}^o(x, v)$ is a read label of value $v \in \mathsf{Val}$ from memory location $x \in \mathsf{Loc}$ with access mode $o \in \mathsf{Mod}$;
- $\texttt{W}^o(x, v)$ is a write label of value $v \in \mathsf{Val}$ into memory location $x \in \mathsf{Loc}$ with access mode $o \in \mathsf{Mod}$;
- $\texttt{F}^o$ is a fence label with mode $o \in \mathsf{Mod}$.

Access mode can be either: *non-atomic* `na`, *relaxed mode* `rlx`, *acquire mode* `acq`, *release* mode `rel`, or *sequentially consistent mode* `sc`.

$$\mathsf{Mod} \triangleq \{\texttt{na}, \texttt{rlx}, \texttt{acq}, \texttt{rel}, \texttt{sc}\}.$$

Access modes are partially ordered by their strength as the following diagram demonstrates:



We also define several *derived* relations on execution graphs.

$\texttt{rb}$	$\triangleq$	$\texttt{rf}^{-1}; \texttt{co}$	reads-before
$\texttt{eco}$	$\triangleq$	$(\texttt{co} \cup \texttt{rf} \cup \texttt{rb})^+$	extended coherence order
$\texttt{sw}$	$\triangleq$	$[\texttt{W}^{\texttt{rel}\sqsubseteq}]; \texttt{rf}; [\texttt{R}^{\texttt{acq}\sqsubseteq}]$	synchronizes-with
$\texttt{hb}$	$\triangleq$	$(\texttt{po} \cup \texttt{sw})^+$	happens-before

In the context of this assignment, we always assume that the `hb` relation is acyclic (*i.e.,* it is a partial order).

Also, recall the definition of the simplified C/C++ memory model, discussed on the lecture. An execution graph G is consistent with respect to this model, if the following condition is met:

$$\texttt{hb}|_{\texttt{loc}} \cup \texttt{rf} \cup \texttt{co} \cup \texttt{rb} \text{ is acyclic relation.}$$

Task 1 (4 points)

Prove that the definition of the simplified C/C++ consistency given above is equivalent to the following condition:

$$\texttt{hb};\texttt{eco} \text{ is irreflexive relation.}$$

Hint: you might find useful the following equality:

$$(r_1 \cup r_2)^+ \triangleq r_1^+ \cup r_1^*;(r_2;r_1^*)^+$$

where r_1 and r_2 are two arbitrary relations.

Proof. **TODO:** □

Task 2 (4 points)

Consider another relation — *writes-before*, defined as follows:

$$\texttt{wb} \triangleq [\texttt{W}];(\texttt{rf}^?;\texttt{hb}|_{\texttt{loc}};(\texttt{rf}^{-1})^?) \setminus \texttt{id};[\texttt{W}]$$

where $\texttt{id} \triangleq \{\langle x, x\rangle \mid x \in \texttt{E}\}$ is an identity relation, and $r^? \triangleq \texttt{id} \cup r$ is a reflexive closure of a relation.

Assuming that the given execution graph does not contain read-modify-write (RMW) events, prove that the definition of the simplified C/C++ consistency is also equivalent to the following condition:

$$\texttt{wb} \text{ is acyclic relation}$$

Hint: try to show that $\texttt{wb} \subseteq \texttt{co}$.

Proof. **TODO:** □

Task 3 (2 points)

Consider the problem of checking whether the given execution graph G is consistent (with respect to the simplified C/C++ consistency model), assuming that the $G.\texttt{co}$ relation is unknown.

Answer the following questions:

1. What graph algoritm can be used to solve this problem (*i.e.,* to check consistency of the graph G).

2. What definition of consistency, among the three equivalent given above, is the most suitable for this algoritm and gives the best time complexity for the overall procedure of consistency checking?

TODO:

References

[1] O. Lahav, N. Giannarakis, and V. Vafeiadis. Taming release-acquire consistency. *ACM SIGPLAN Notices*, 51(1):649–662, 2016.