The Temporal Logic of Programs

Amir Pnueli (1977)

The Point of this Paper

- Two formal systems giving a sound basis for temporal reasoning about correctness of sequential *and* concurrent programs
    - Intermittent assertions
    - Tense logic system K_b
- *Correctness* of a program is reduced to two main concepts
    - **Invariance**: property holds throughout
    - **Eventuality**: temporal implication
- Prior work focused on functional programs only, ignoring OS/real-time systems where **halting is abnormal behavior**

Systems and Programs

General Framework

A **system** is

$$(S, R, s_o)$$

where

- S: the (possibly infinte) set of states $\{s_i\}$
- R: transition relation between state and successors, $R \subseteq S \times S$
- $s_o \in S$: initial state

Execution is the sequence

$$\sigma = s_o, s_1, \ldots$$

where for each $i \geq 0, R(s_i, s_{i+1})$ holds

Sequential Programs

The **state component** s of deterministic sequential programs is

$$s = (\pi, u)$$

where

- π: the **control** component taking as values *program locations* $L = \{l_0, l_1, ..., l_n\}$
- u: the **data** component

The **transition relation** R is composed of

- $N(\pi, u)$: next-location function
- $T(\pi, u)$: data transformation function

such that

$$R((\pi, u), (\pi', u')) \Leftrightarrow \pi' = N(\pi, u) \wedge u' = T(\pi, u)$$

Concurrent Programs

The **state component** s of concurrent programs allows more than one control component

$$s = (\pi_1, ..., \pi_n; u)$$

where the range for each π_i is the program for the i^{th} processor (scheduling is nondeterministic).

The **transition relation** R is

$$R((\pi_1, ...\pi_n; u), (\pi_i', ...\pi_n'; u')) \Leftrightarrow$$

$$\exists i, 1 \leq i \leq n : (\pi_1', ...\pi_n') = (\pi_1, ...\pi_{i-1}, N_i(\pi_i, u), \pi_{i+1}, ...\pi_n),$$

$$u' = T_i(\pi_i, u)$$

Specifications

Specifications about Time

Establish facts about development of properties $q(s)$ in time, where

- $q(\pi_1, ...\pi_n; u)$ is a relation between data values and location of all proccessor pointers

When t ranges over time, we say that $H(t, q) \equiv q(s_t)$

Single Time Instance Specifications

Invariance

Defining set of accessible states as

$$X = \{s | R^*(s_0, s)\}$$

a predicate $p(s)$ is **invariant** if

$$\forall s \in X, p(s) \equiv \forall t, H(t, p)$$

See in paper:

- Partial Correctness
- Mutual Exclusion
- Deadlock Freedom

Two Time Instances

Eventuality (Temporal Implication)

Write $\phi \leadsto \psi$ for

$$\forall t_1 \exists t_2, t_2 \geq t_1 \wedge H(t_1, \phi) \supset H(t_2, \psi)$$

meaning for every execution $\sigma = s_0, s_1, ...$, whenever there exists an s_i such that $\phi(s_i)$, there must exist a later $s_j, j \geq i$ such that $\psi(s_j)$.

See in paper:

- Total correctness
- Accessibility
- Responsiveness
- Fairness

Proof Principles

- Principle of Computational Induction (P1)
- Well-Founded Sets (P2)
- An Axiomatic System over Intermittent Assertions (ER)

They use ER to derive eventualities, showing it is sound and complete for proving any property of the form $\phi \rightsquigarrow \psi$.

Let's **go to the paper**.