Transition systems.

1 Transition systems.

Definition 1. A transition system is a tuple

$$TS = (S, \mathrm{Act}, \to, I, \mathrm{AP}, L)$$

where

- ▷ S is the set of *states*;
- ▷ Act is the set of *actions*;
- ▷ $\to \subseteq S \times \mathrm{Act} \times S$ the *transition relation*;
- ▷ $I \subseteq S$ the set of *initial states*;
- ▷ AP is the set of *atomic propositions*;
- ▷ $L : S \to \wp(\mathrm{AP}) \cong \mathbf{2}^{\mathrm{AP}}$ is the *state labelling function.*

We will write $s \xrightarrow{\alpha} s'$ when $(s, \alpha, s') \in \to$.

Example 1 (Beverage Vending Machine, BVM). We can model a beverage vending machine using a diagram like in figure 1. Here we have that:

- ▷ $S = \{\mathsf{pay}, \mathsf{select}, \mathsf{soda}, \mathsf{beer}\}$,
- ▷ $I = \{\mathsf{pay}\}$,
- ▷ $\mathrm{Act} = \{\mathtt{ic}, \tau, \mathsf{gb}, \mathsf{gs}\}$.[1]

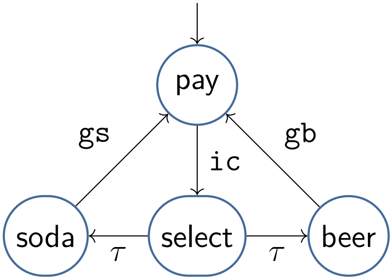

Figure 1 | *Transition system for the BVM*

We can define the labels:

$$L(\mathsf{pay}) = \emptyset \quad L(\mathsf{soda}) = L(\mathsf{beer}) = \{\mathsf{paid}, \mathsf{drink}\} \quad L(\mathsf{select}) = \{\mathsf{paid}\},$$

with $\mathrm{AP} = \{\mathsf{paid}, \mathsf{drink}\}$.

2 Program graphs.

The goal is to represent the evaluation of a program.

Definition 2 (Typed variables). ▷ A set Var of *variables*.

- ▷ For each variable $x \in \mathrm{Var}$, consider a set $\mathrm{Dom}(x)$.
- ▷ Given $TV = (\mathrm{Var}, (\mathrm{Dom}(x))_{x\in\mathrm{Var}})$, we define

$$\mathrm{Eval}(TV) = \prod_{x\in\mathrm{Var}} \mathrm{Dom}(x),$$

the set of valuations of the form $\eta : x \in \mathrm{Var} \mapsto \eta(x) \in \mathrm{Dom}(x)$.

[1]The meaning of the actions are the following: `ic` means *insert coin*, `gb` means *get beer* and `gs` for *get soda*.

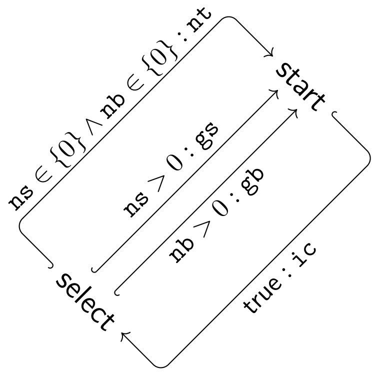

Figure 2 | *BVM as a program graph*

Definition 3 (Program graph). A *program graph* is a tuple

$$PG = (\mathrm{Loc}, \mathrm{Act}, \mathrm{Effect}, \hookrightarrow, \mathrm{Loc}_0, g_0),$$

where

- Loc is the set of *locations* (lines of codes);
- *Act* is the set of *actions*;
- $\mathrm{Effect} : \mathrm{Act} \times \mathrm{Eval}(TV) \to \mathrm{Eval}(TV)$;
- $\hookrightarrow \subseteq \mathrm{Loc} \times \mathrm{Conditions} \times \mathrm{Act} \times \mathrm{Loc}$ where conditions are propositional formula built from atoms of the forms "$x \in D$" for some variable x and some set $D \subseteq \mathrm{Dom}(x)$;
- $\mathrm{Loc}_0 \subseteq \mathrm{Loc}$ the set of initial locations;
- g_0 is the *initial condition.*

We will write $\ell \overset{g:\alpha}{\hookrightarrow} \ell'$ for $(\ell, g, \alpha, \ell') \in \hookrightarrow$.

Example 2 (BVM as a program graph). In figure 2, we use

- $\mathrm{Loc} = \{\mathsf{start}, \mathsf{select}\}$;

- $\text{Var} = \{\texttt{ns}, \texttt{nb}\}$;
- $\text{Act} = \{\texttt{ic}, \texttt{nt}, \texttt{gs}, \texttt{gb}, \texttt{refill}\}$;
- $\text{Loc}_0 = \{\textsf{start}\}$;
- $g_0 = \texttt{ns} \in \{\max\} \wedge \texttt{nb} \in \{\max\}$
- $$\begin{aligned}\text{Effect} : \text{Act} \times \text{Eval}(TV) &\longrightarrow \text{Eval}(TV)\\ (\texttt{refill}, \eta) &\longmapsto [\texttt{ns} \mapsto \max, \texttt{nb} \mapsto \max]\\ (\texttt{gs}, \eta) &\longmapsto \eta[\texttt{ns} \mapsto \eta(\texttt{ns}) - 1]\\ (\texttt{gb}, \eta) &\longmapsto \eta[\texttt{nb} \mapsto \eta(\texttt{nb}) - 1]\end{aligned}$$

3 Transition system of a program graph.

Definition 4. Given TV and PG a program graph, we define

$$TS(PG) := (S, \text{Act}, \to, I, \text{AP}, L)$$

where

- $S = \text{Loc} \times \text{Eval}(TV)$;
- $\text{AP} = \text{Loc} \cup \text{Conditions}$;
- $I = \{(\ell_0, \eta) \mid \ell_0 \in \text{Loc}_0, \eta \models g_0\}$;
- $\to$ is defined by:

$$\frac{\ell \xhookrightarrow{g:\alpha} \ell' \qquad \eta \models g}{(\ell, \eta) \xrightarrow{\alpha} (\ell', \text{Effect}(\alpha, \eta))},$$

- and $L(\ell, \eta) = \{\ell\} \cup \{g \mid \eta \models g\}$.

> **Example 3.** The BVM program graph example seen in the previous example can be transformed as a transition system thanks to the previous definition; it is shown in figure 3. To simplify, we assume $\max = 1$.

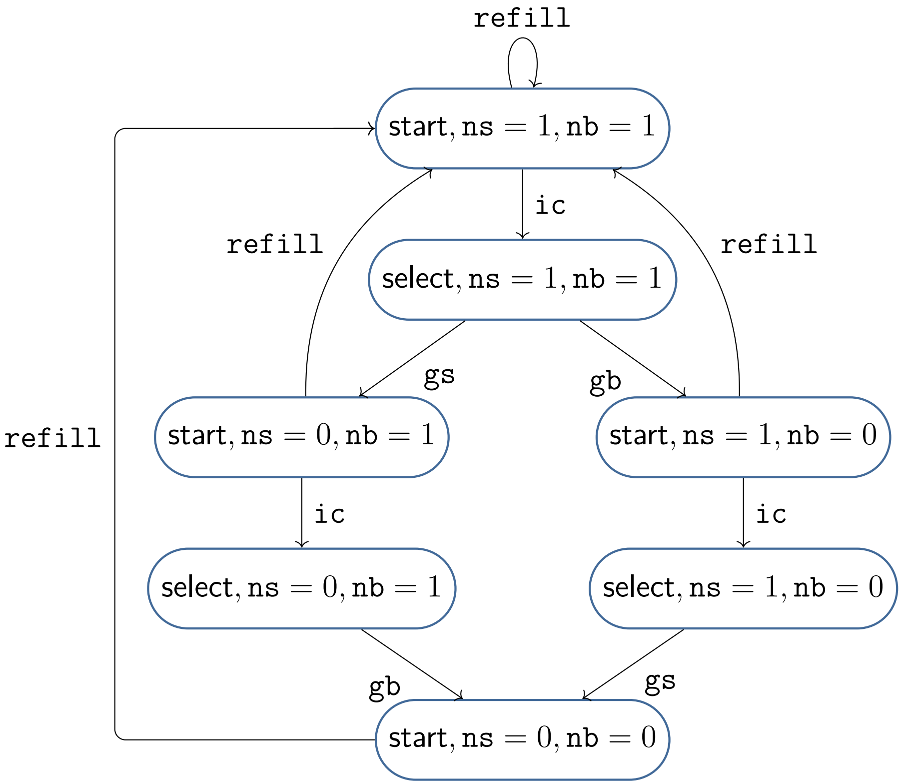

Figure 3 | *Transition system of the PVM program graph*