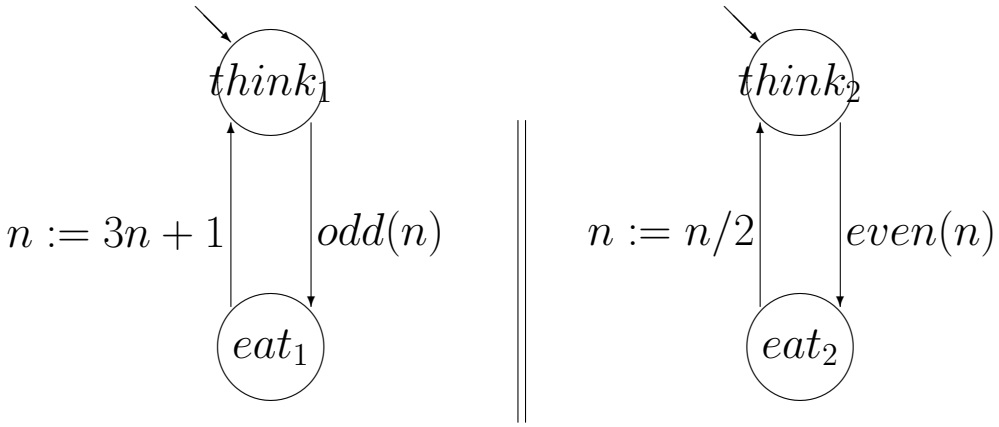
Abstractions

- State-spaces often get very large, with many variables
- Often, many of the variables are irrelevant for the analysis to be performed,
- or only some particular property of the variable is relevant.
- **Idea:** Create a simpler model, which "abstracts away" the irrelevant detail in the original model.
- Ideally, the "abstracting away" should be performed automatically, by a "compiler".
- The abstract model can be analyzed by a model-checker.

Example:

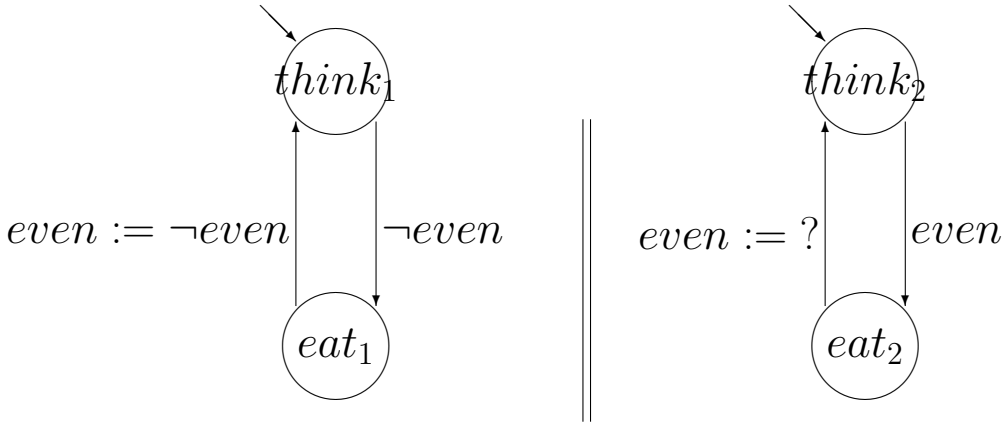
Action System *Dining Mathematicians*
declare n : integer
initially $n \geq 1$



end

If we are only interested in control:

Action System *Abstract Mathematicians*
declare *even* : boolean
initially



end

How to relate the abstract to the more detailed?

- Let $\langle S, S^0, \rightarrow \rangle$ be the detailed transition system,
  Let $\langle T, T^0, \rightarrow \rangle$ be the abstract transition system,
- **Idea:** Each abstract state represents a set of concrete states.
- **Formally:** This is represented by a *concretization function* $\gamma : T \mapsto \mathcal{P}(S)$, which maps each abstract state to a set of "concrete" states.
- We also want to "compute" on the set of abstract values. E.g., we want to take "union", "intersection", and so on.
- Define an ordering $\sqsubseteq$ on abstract states by

$$t_1 \sqsubseteq t_2 \quad \equiv \quad \gamma(t_1) \subseteq \gamma(t_2)$$

- For each set $F \subseteq S$, define a "best approximation" $\alpha(F) \in T$, such that

$$\begin{array}{ll} \alpha(\gamma(t)) = t & \text{for each } t \in T \\ \gamma(\alpha(F)) \supseteq F & \text{for each } F \subseteq S \end{array}$$

  We say that (α, γ) forms a *Galois Insertion* from $(\mathcal{P}(S), \subseteq)$ to $(T, \sqsubseteq)$.

Approximating the Transition Relation

- The relation $\rightarrow$ is a *safe abstraction* of the relation $\rightarrow$ between concrete states if

  whenever $s \in \gamma(t)$ and $s \longrightarrow s'$

  there is t' such that $s' \in \gamma(t')$ and $t \longrightarrow t'$

- We say that $\langle T, T^0, \rightarrow \rangle$ is a safe abstraction of $\langle S, S^0, \rightarrow \rangle$ if
  - $\rightarrow$ is a safe abstraction of $\rightarrow$, and
  - $\forall s^0 \in S^0.\ \exists t^0 \in T^0.\ s^0 \in \gamma t^0$

- **Theorem:** If $\langle T, T^0, \rightarrow \rangle$ is a safe abstraction of $\langle S, S^0, \rightarrow \rangle$, then for each computation

  $$s^0 \ \ s^1 \ \ s^2 \ \ s^3 \ \ s^4 \ \ s^5 \ \ \ldots$$

  of $\langle S, S^0, \rightarrow \rangle$, there is a computation

  $$t^0 \ \ t^1 \ \ t^2 \ \ t^3 \ \ t^4 \ \ t^5 \ \ \ldots$$

  of $\langle T, T^0, \rightarrow \rangle$ such that

  $$s_i \in \gamma(t_i)$$

  for each i.

Relating Concrete and Abstract Behaviors and their Properties

For a system, and a temporal property ϕ, we want

$$\langle T, T^0, \rightarrow \rangle \models \phi \qquad \text{implies} \qquad \langle S, S^0, \rightarrow \rangle \models \phi$$

This can work, if we define

- For "simple" state formulas p:

  $$t \models p \qquad \text{iff} \qquad \forall s \in \gamma(t).\ s \models p$$

  Note that "the law of excluded middle" need not hold!

- Extend the above definition to properties of behaviors in the natural way.

- **Theorem:**

  $$\langle T, T^0, \rightarrow \rangle \models \phi \qquad \text{implies} \qquad \langle S, S^0, \rightarrow \rangle \models \phi$$

Induced Operations and Transitions

Often, we want to compute an abstraction directly, based on the syntax.

- Assume that the original model has a set of actions (e.g., corresponding to "basic statements")
- Assume that states are approximated in an abstract domain, as before:
- Approximate each action $\mathcal{A}$ by its *induced action*

$$t \longrightarrow \alpha(post(\mathcal{A}, \gamma(t)))$$

- It is, more precisely, enough that we Approximate each action $\mathcal{A}$ by an abstract action $\mathcal{B}$ such that

$$\alpha(post(\mathcal{A}, \gamma(t))) = post(\mathcal{B}, t)$$

- **Example:** Let the set of concrete states be $\mathcal{N}$. Let the abstract domain be $\mathcal{P}(\{even, odd\})$. Define

$$\begin{array}{lcl} \gamma(even) & = & \{0, 2, 4, \ldots\} \\ \gamma(odd) & = & \{1, 3, 5, \ldots\} \\ \gamma(\{even, odd\}) & = & \mathcal{N} \\ \gamma(\emptyset) & = & \emptyset \end{array}$$

- **Induced Operations:** can be tabled as follows:

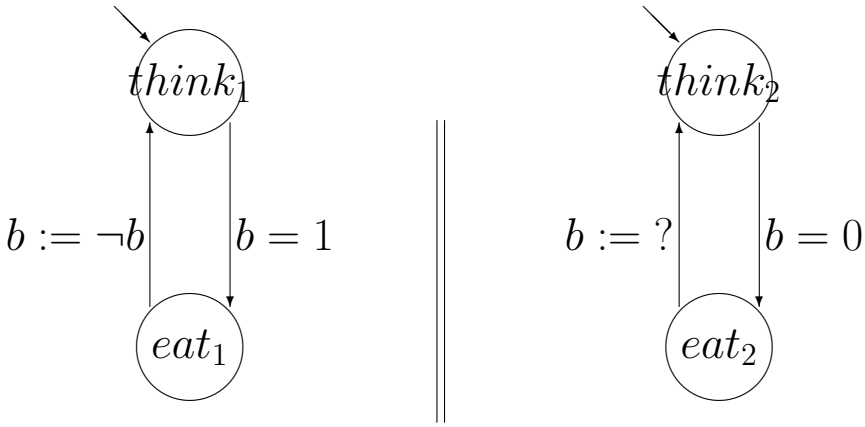
Operator	**Result on** *even*	**Result on** *odd*
$\cdot + 1$	*odd*	*even*
$3 * \cdot$	*even*	*odd*
$\cdot/2$	$\{even, odd\}$	$\emptyset$
$even(\cdot)$	*true*	*false*
$odd(\cdot)$	*false*	*true*

The table is extende to sets by

$$op(t_1 \sqcup t_2) = op(t_1) \sqcup op(t_2)$$

Abstract System

Action System *Dining Mathematicians*
declare b : bit
initially *true*



end

Problem: Check whether

$$\neg(at\ eat_1 \wedge at\ eat_2)$$

is invariant.

What about Liveness and Fairness?