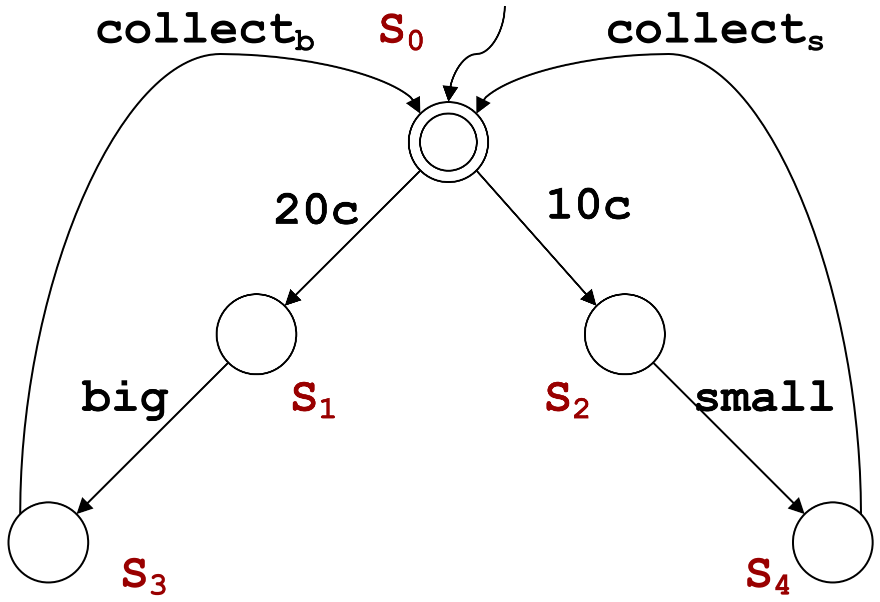
Transition Systems

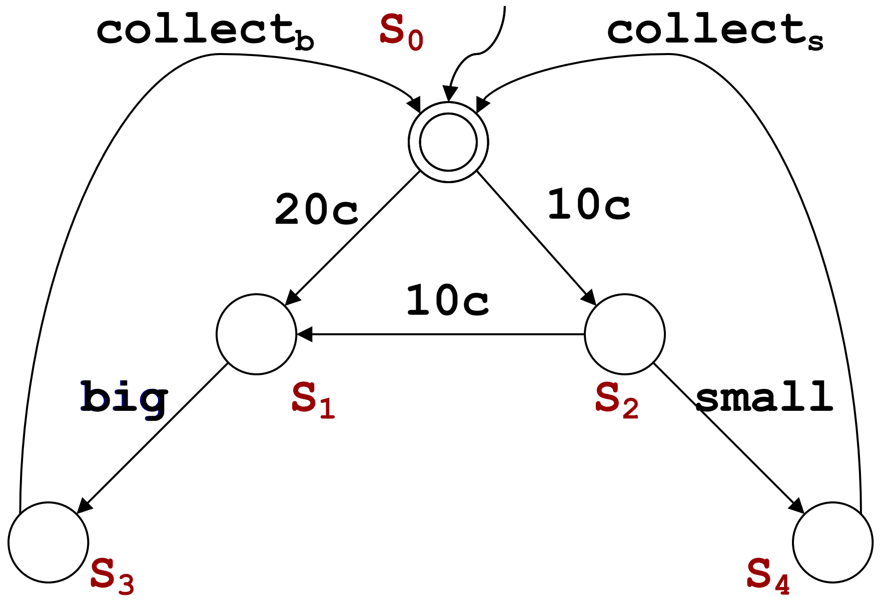
Giuseppe De Giacomo

Transition Systems

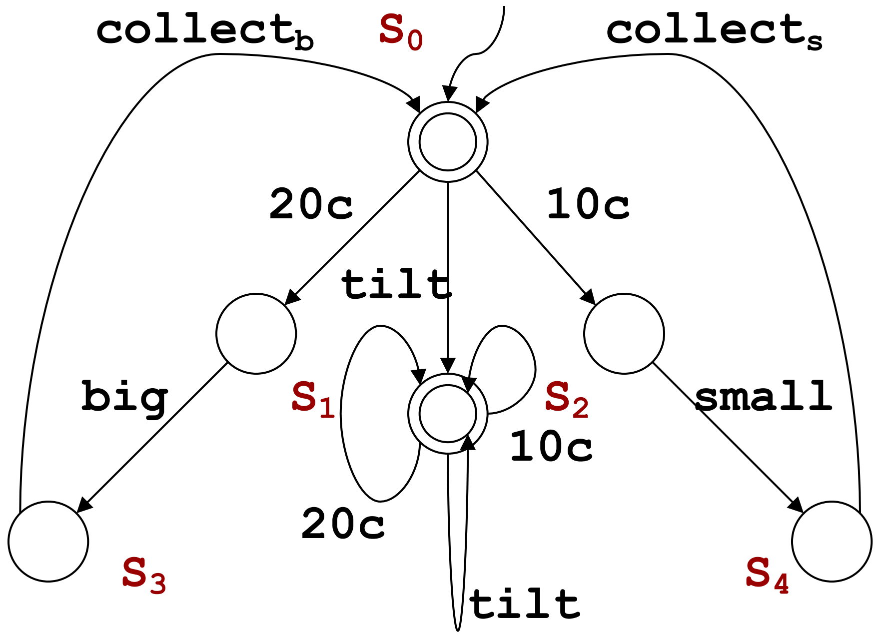
Concentrating on behaviors: Vending Machine



Concentrating on behaviors: Another Vending Machine

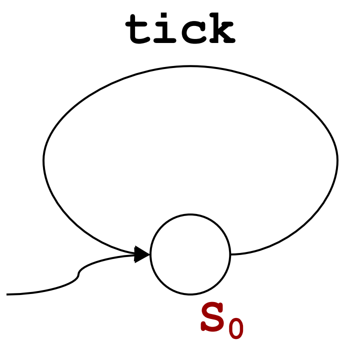


Concentrating on behaviors: Vending Machine with Tilt



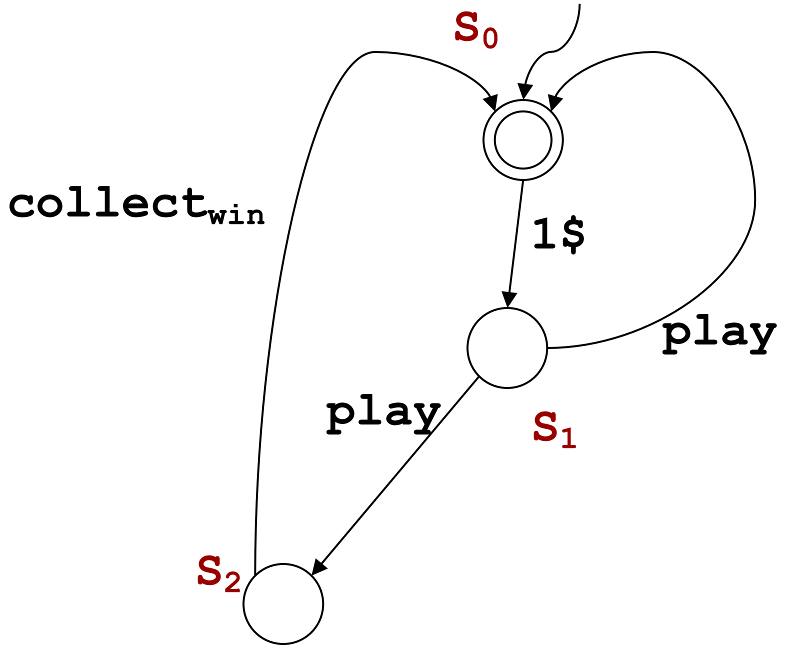
Example (Clock)

TS may describe (legal) nonterminating processes

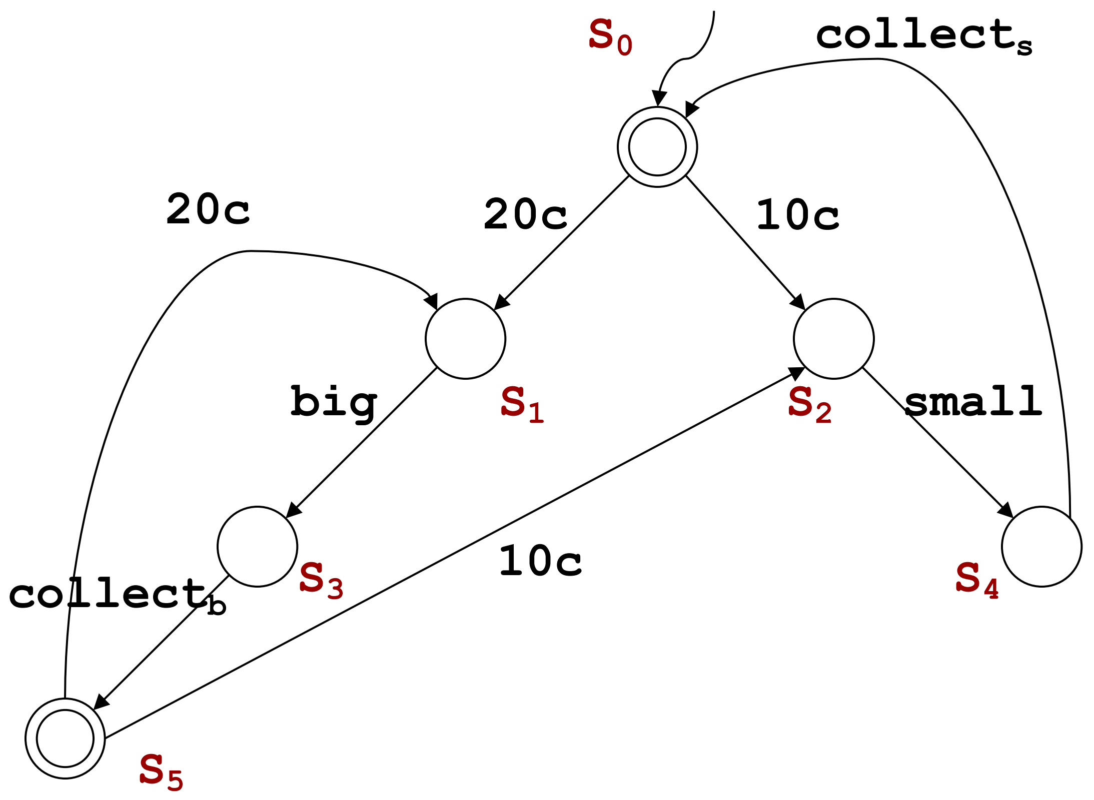


Example (Slot Machine)

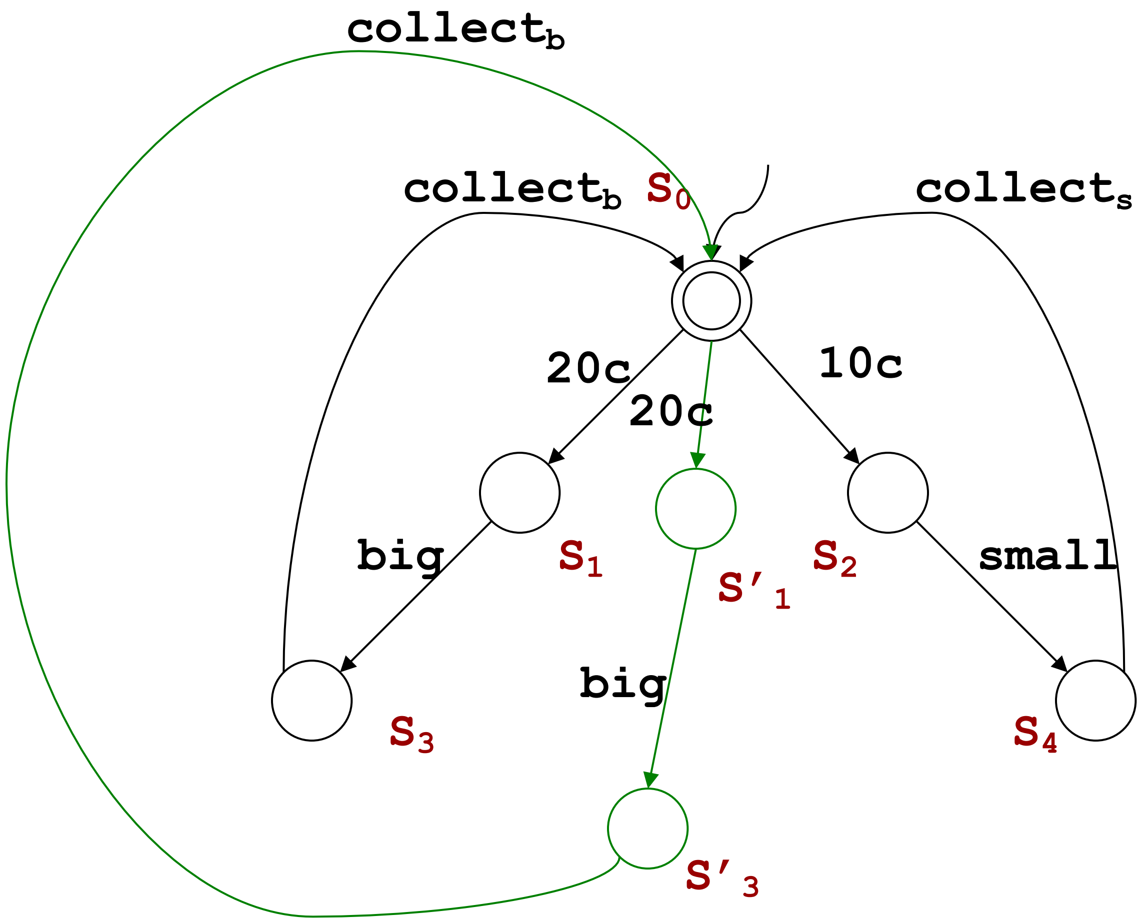
Nondeterministic transitions express
choice that is **not** under the **control** of clients



Example (Vending Machine - Variant 1)



Example (Vending Machine - Variant 2)



Transition Systems

- A transition system TS is a tuple $T = < A, S, S^0, \delta, F>$ where:
  - A is the set of actions
  - S is the set of states
  - $S^0 \subseteq S$ is the set of initial states
  - $\delta \subseteq S \times A \times S$ is the transition relation
  - $F \subseteq S$ is the set of final states

(c.f. Kripke Structure)

- Variants:
  - No initial states
  - Single initial state
  - Deterministic actions
  - States labeled by propositions other than Final/$\neg$Final

Bisimulation and Bisimilarity

Bisimulation

Intuition:

Two (states of two) transition systems are bisimilar if they have the same behavior.

In the sense that:

- *Locally they (the two **states**) look indistinguishable*
- *Every **action** that can be done on one of them can also be done on the other remaining indistinguishable*

Bisimulation

- A binary relation R is a **bisimulation** iff:

  $(s,t) \in R$ implies that
  - s is *final* iff t is *final*
  - for all actions a
    - if $s \rightarrow_a s'$ then $\exists\, t'\,.\, t \rightarrow_a t'$ and $(s',t') \in R$
    - if $t \rightarrow_a t'$ then $\exists\, s'\,.\, s \rightarrow_a s'$ and $(s',t') \in R$

- A state s_0 of transition system S is **bisimilar**, or simply **equivalent,** to a state t_0 of transition system T iff there **exists** a **bisimulation** between the initial states s_0 and t_0.

- Notably
  - **bisimilarity** is a bisimulation
  - **bisimilarity** is the largest bisimulation

Note it is a co-inductive definition!

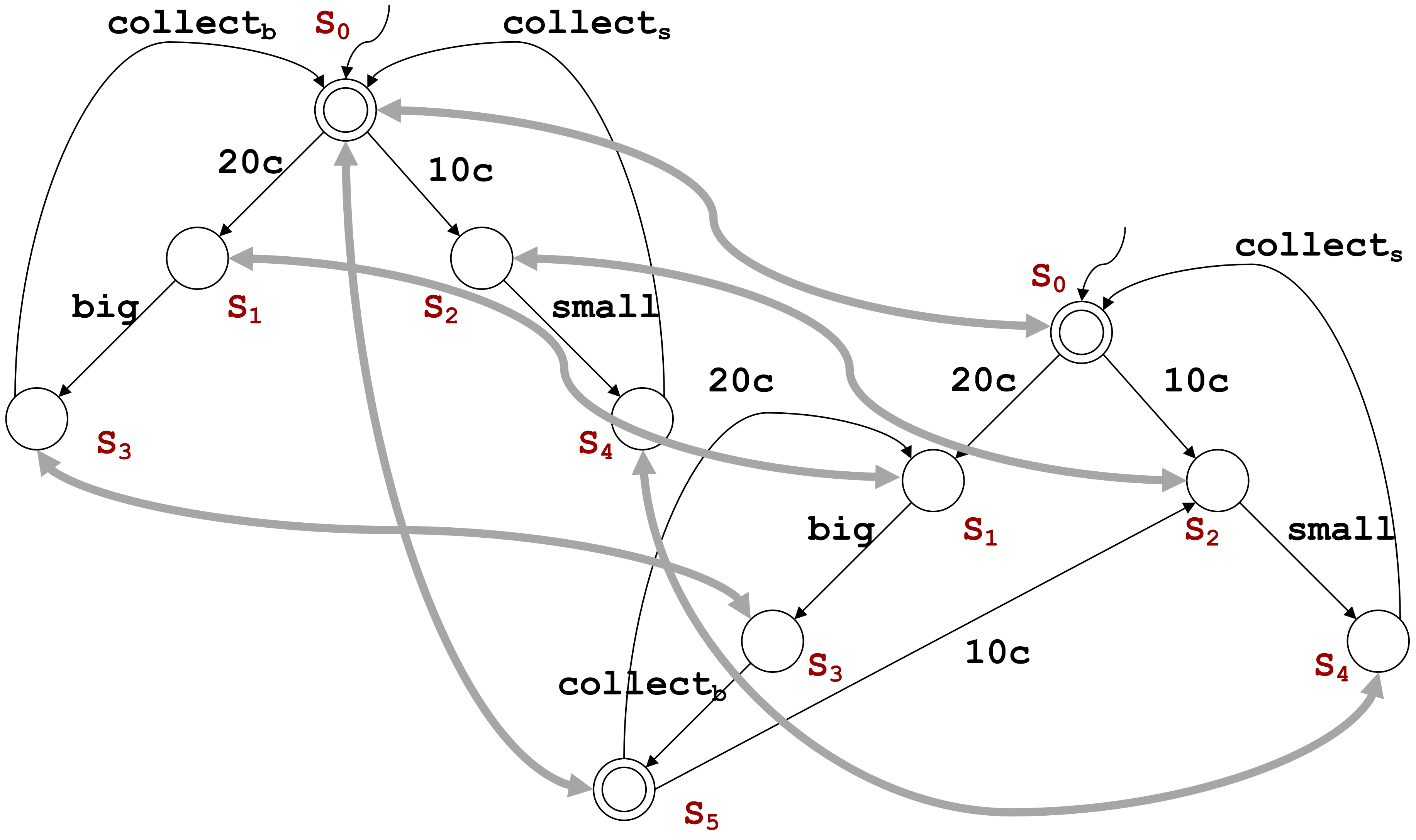
Computing Bisimulation on Finite Transition Systems

Algorithm ComputingBisimulation
Input: transition system TS_S = < A, S, S^0, δ_S, F_S> and
transition system TS_T = < A, T, T^0, δ_T, F_T>
Output: the **bisimilarity** relation (the largest bisimulation)

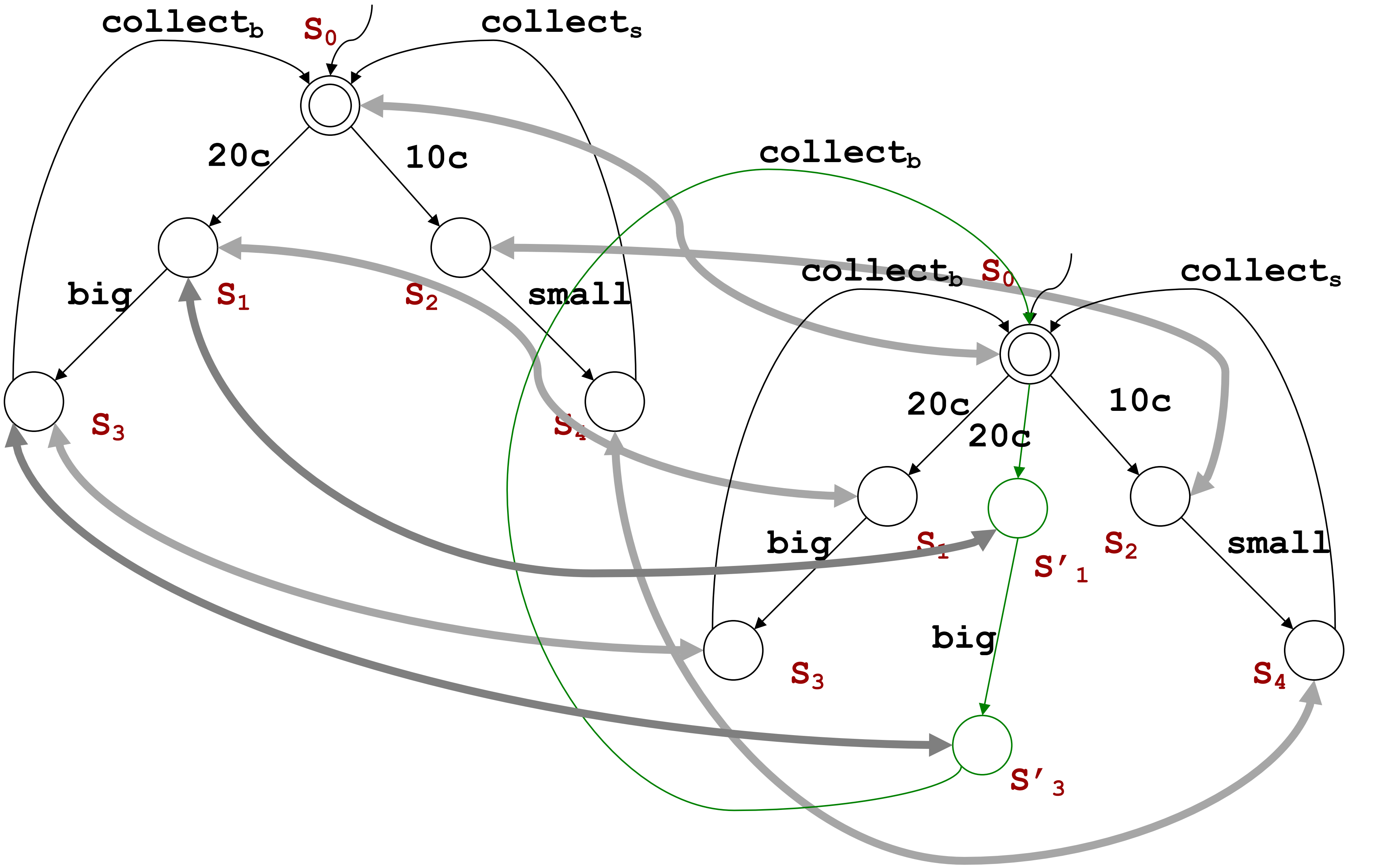
Body
R = S $\times$ T
R' = R - {(s,t) | $\neg$(s $\in F_S$ ´ t $\in F_T$)}
while (R ≠ R') {
R := R'
R' := R' - ({(s,t) | $\exists$ s',a. s $\rightarrow_a$ s' $\wedge$ $\neg\exists$ t' . t $\rightarrow_a$ t' $\wedge$ (s',t') $\in$ R' }
{(s,t) | $\exists$ t',a. t $\rightarrow_a$ t' $\wedge$ $\neg\exists$ s' . s $\rightarrow_a$ s' $\wedge$ (s',t') $\in$ R' })
}
return R'
Ydob

*This algorithm is based on computing iteratively fixpoint approximates for the **greatest fixpoint**, starting from the total set (SxT).*

Example of Bisimulation



Example of Bisimulation

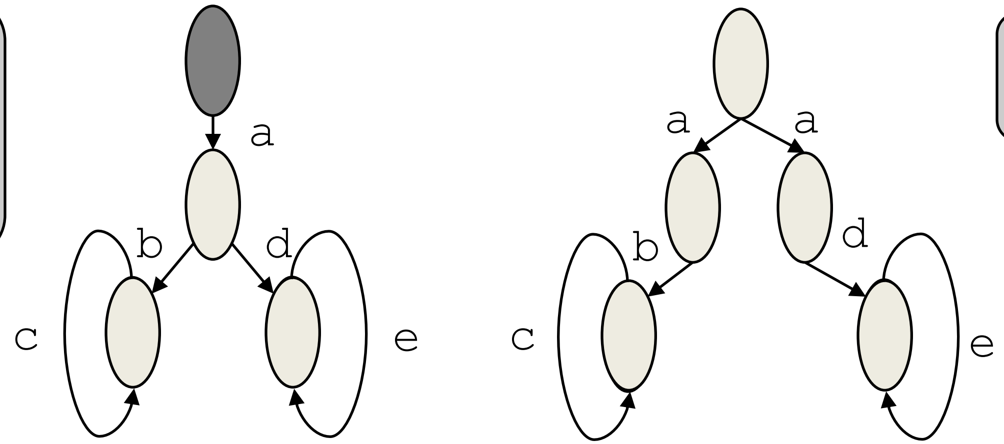


Automata vs.Transition Systems

- Automata
  - define sets of runs (or traces or strings): (finite) length sequences of actions
- TSs
  - … but I can be interested also in the alternatives “encountered” during runs, as they represent client’s “choice points”

As automata they recognize the same language:

abc* + ade*



Different as TSs