Verification of Real Time Systems - CS5270 6th lecture

Hugh Anderson

National University of Singapore
School of Computing

February, 2007

A warning...

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...

2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA

3. Reducing complexity
   - Quotienting

The mid semester test

During next timetabled class

- The mid semester test is on March 1, 2007
- During the lecture, in this room.
- 1 hour
- Similar to last year's test (handed out in class)

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

Assignment 2

Assignment number 2:

- Not ready yet - will try to put up over the weekend

Outline

The immediate road map

After completing scheduling, next 2/3 weeks have three topics:

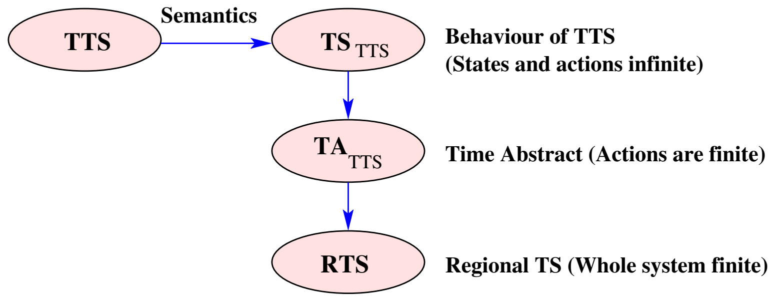
- **TS: State transition systems**
  - some definitions
  - parallel composition
- **TTS: Timed transition systems**
  - formal definition
  - parallel composition

  - Reduction of a TTS (which has possibly infinite states and actions) to a finite TS by quotienting? (takes time)
- **Efficiency in TTS**
  - Regions
  - zones

Outline

The process...

Three steps...



Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

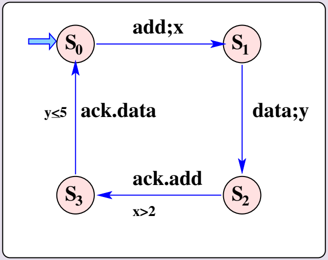
From TTS to TS$_{\mathrm{TTS}}$

The reduction steps...

- $\mathrm{TTS} = (S, S_{\mathrm{in}}, \mathrm{Act}, X, I, \rightarrow)$
- $\mathrm{TS}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act} \cup \mathbb{R}, \Longrightarrow)$

Representing a TTS with TS

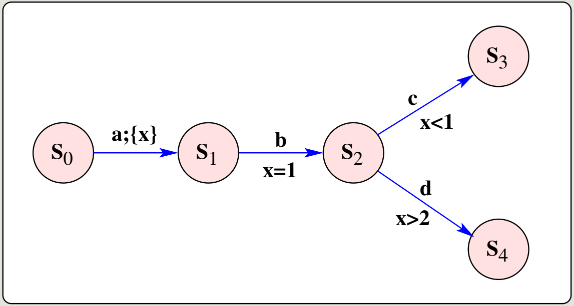
Behaviour of TTS linked with time



- The transition system $\mathrm{TS}_{\mathrm{TTS}}$ works on (possibly infinite) sets of states $\mathcal{S}$ of the form $S \times V$, where V is a *valuation* (the current values of each clock variable).
- In the figure, $(s_1, (2, 5))$ is an example of a state in $\mathcal{S}$.

Example of states/behaviours

Consider this TTS:



- $(S_1, 0)$ $(S_2, 1.8)$ (S_4, π) are timed-states (t-states).
- $(S_3, 5)$ is a t-state but not reachable.

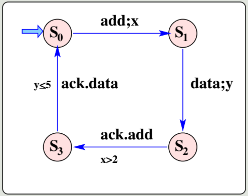
Representing a TTS with TS

Behaviour of TTS linked with time

- Given a timed transition system $\mathrm{TTS} = (S, S_{\mathrm{in}}, \mathrm{Act}, X, I, \rightarrow)$, we can derive the associated transition system $\mathrm{TS}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act} \cup \mathbb{R}, \Longrightarrow)$ where
  - $\mathcal{S}$ is a (possibly infinite) set of pairs $S \times V$,
  - $\mathcal{S}_0$ is $S_0 \times \{V_0\}$,
  - V are the valuations of the clock variables ($V : X \rightarrow \mathbb{R}$), and finally
  - $\Longrightarrow \subseteq \mathcal{S} \times (\mathrm{Act} \cup \mathbb{R}) \times \mathcal{S}$.

Example of states/behaviours

Consider this TTS:



- $(S_1, (2, 5))$ is a state: $(S_1, V) : V(x) = 2 \land V(y) = 5$
- $(S_2, (15, 0))$ is a state. $(S_1, V') : V'(x) = 15 \land V(y) = 0$

A possible trace is

$(S_0, (0, 0))\, 1.6\, (S_0, (1.6, 1.6))\, \text{add}\, (S_1, (0, 1.6))\, 2\, (S_1, (2, 3.6))\ldots$

Types of transitions

We have two types of transitions:

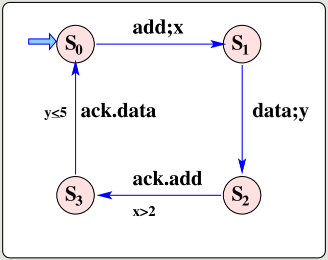
1. **Time passing move:** $(s, V) \stackrel{\delta}{\Longrightarrow} (s, V + \delta)$, with $\delta \geq 0$
2. **Action move:** $(s, V) \stackrel{a}{\Longrightarrow} (s', V')$

Two consecutive time passing moves can be amalgamated into one time passing move.

For example $(s_0, (0, 0))\, 0.6\, (s_0, (0.6, 0.6))\, 0.6\, (s_0, (1.2, 1.2))$ can be amalgamated into $(s_0, (0, 0))\, 1.2\, (s_0, (1.2, 1.2))$.

Time-passing move

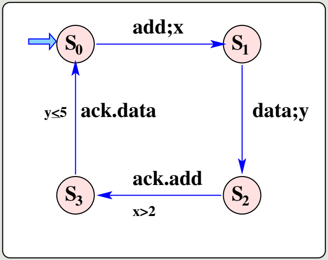
Example 1 - Consider this TTS:



- Is this a time passing move? $(S_1, (0, 5)) \stackrel{1}{\Longrightarrow} (S_1, (1, 6))$

Time-passing move

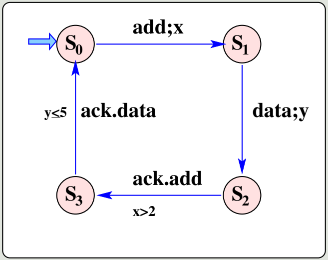
Example 2 - Consider this TTS:



- Is this a time passing move? $(S_1, (0,5)) \xrightarrow{0} (S_1, (0,5))$

Time-passing move

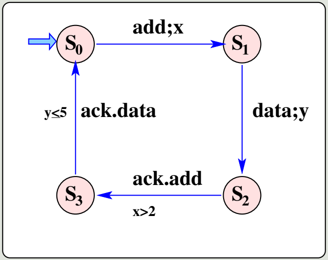
Example 3 - Consider this TTS:



- Is this a time passing move? $(S_1, (0, 5)) \xrightarrow{2} (S_1, (2, 7.7))$

Action move

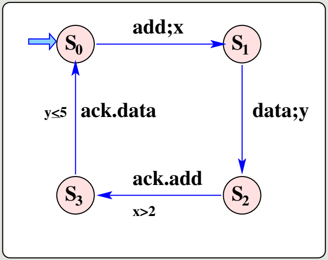
Example 1 - Consider this TTS:



- Is this a possible transition? $(S_0, (3,3)) \xrightarrow{\text{add}} (S_1, (0,3))$

Action move

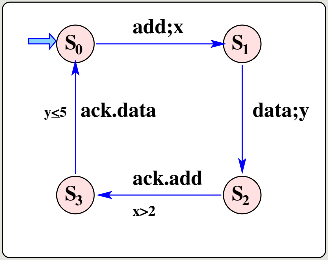
Example 2 - Consider this TTS:



- Is this a possible transition? $(S_0, (3,3)) \xrightarrow{add} (S_3, (0,3))$

Action move

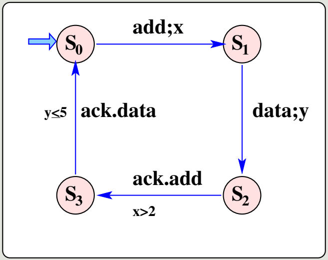
Example 3 - Consider this TTS:



- Is this a possible transition? $(S_0, (3,3)) \overset{\text{add}}{\Longrightarrow} (S_1, (0,4))$

Action move

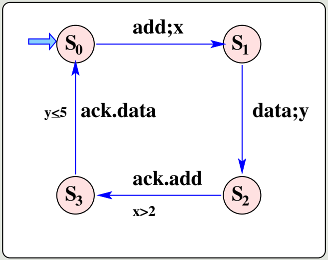
Example 4 - Consider this TTS:



- Is this a possible transition? $(S_0, (0,0)) \stackrel{add}{\Longrightarrow} (S_1, (0,0))$

Action move

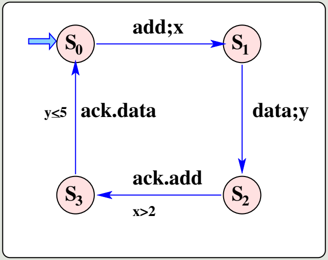
Example 5 - Consider this TTS:



- Is thispossible? $(S_0, (0,0)) \overset{add}{\Longrightarrow} (S_1, (0,0) \overset{add}{\Longrightarrow} (S_2, (0,0))$

Action move

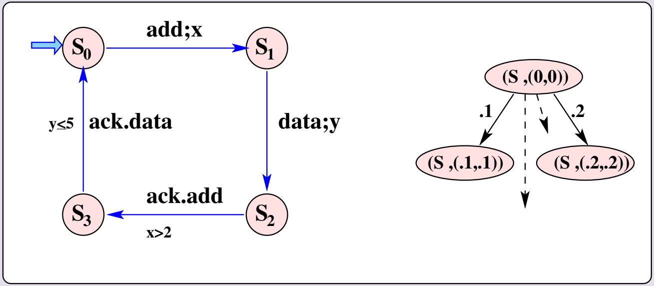
Example 6 - Consider this TTS:



- Is this a possible transition? $(S_2, (3, 2)) \xrightarrow{\text{ack.add}} (S_3, (3, 2))$
- Is this a possible transition? $(S_3, (5, 5)) \xrightarrow{\text{ack.data}} (S_0, (5, 5))$

TS is infinite!

Consider the number of states and transitions in the TS



- $\mathrm{TS}_{\mathrm{TTS}}$ will have (uncountably) infinite number of states and transitions.

The behaviour of the TTS

Defined in terms of TS:

$\mathrm{TS_{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act} \cup \mathbb{R}, \Longrightarrow)$ represents the behaviour of $\mathrm{TTS} = (S, S_{\mathrm{in}}, \mathrm{Act}, X, I, \rightarrow)$ in terms of the reachability of states, for $(s, V) \overset{\delta}{\Longrightarrow} (s, V + \delta)$ and $(s, V) \overset{a}{\Longrightarrow} (s', V')$ transitions, provided

$$s \xrightarrow[g]{a;y} s'$$

such that the following conditions are true:

$$V'(x) = \begin{cases} 0 & \text{if } x \in X \\ V(x) & \text{otherwise} \end{cases}$$

$$V \text{ satisfies } g, \text{ the guard for the transition.}$$

The behaviour of the TTS

Runs and computations

In the transition system $\mathrm{TS}_{\mathrm{TTS}}$ we can record runs as for transition systems:

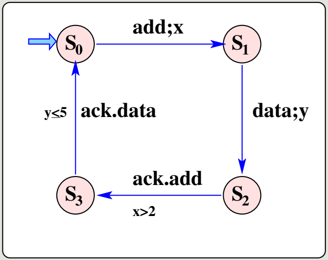
$$(s_0, V_0) \xrightarrow{\delta_0} (s_0, V_0') \xrightarrow{a} (s_1, V_1) \xrightarrow{\delta_1} (s_1, V_1') \xrightarrow{a_1} (s_2, V_2)$$

and $s \in S$ is reachable if and only if there is a computation $(s_0, V_0) \xrightarrow{*} (s_n, V_n)$ in $\mathrm{TS}_{\mathrm{TTS}}$ such that $s_n = s$.

Definition: $s \in S$ is reachable in a TTS if and only if there exists an $(s, V) \in \mathcal{S}$ such that (s, V) is reachable in $\mathrm{TS}_{\mathrm{TTS}}$.

Timed computation

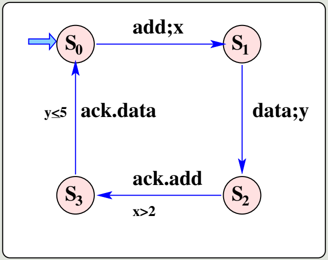
Example 1 - Consider this TTS:



- Is this a timed computation? $(\mathrm{add}, 1)\ (\mathrm{data}, 10)\ (\mathrm{ack.add}, 3)$

Timed computation

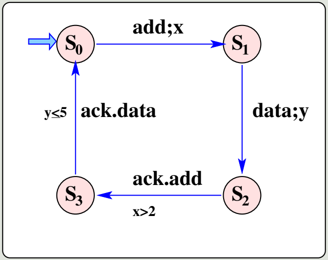
Example 2 - Consider this TTS:



- Is this a timed computation? $(\mathrm{add}, 1)\ (\mathrm{ack.add}, 3)\ (\mathrm{data}, 10)$

Timed computation

Example 3 - Consider this TTS:



- Is this a timed computation? $(\text{add}, 1)\ (\text{data}, 1)\ (\text{ack.add}, 10)$

Outline

1. Administration
   - Mid-semester test!
   - Assignment 2
   - The road map...
2. Reduction of TTS
   - Overview of reduction of TTS
   - From TTS to TS
   - From TS to TA
3. Reducing complexity
   - Quotienting

From TTS to $\mathrm{TS}_{\mathrm{TTS}}$ to $\mathrm{TA}_{\mathrm{TTS}}$

The reduction steps...

- $\mathrm{TTS} = (S, S_{\mathrm{in}}, \mathrm{Act}, X, I, \rightarrow)$
- $\mathrm{TS}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act} \cup \mathbb{R}, \Longrightarrow)$
- $\mathrm{TA}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act}, \rightsquigarrow)$

From $\mathrm{TS}_{\mathrm{TTS}}$ to $\mathrm{TA}_{\mathrm{TTS}}$

The next reduction...

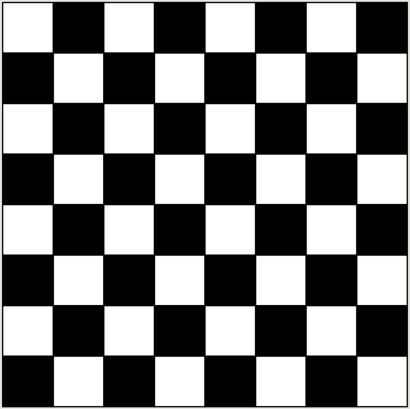
- The *behaviour* of TTS can be represented by the transition system $\mathrm{TS}_{\mathrm{TTS}}$.
- Next step is to look at the reduction from $\mathrm{TS}_{\mathrm{TTS}}$ to the time-abstract transition system $\mathrm{TA}_{\mathrm{TTS}}$, which has only action moves, and not time-passing moves.

We can derive a time-abstract transition system $\mathrm{TA}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act}, \rightsquigarrow)$ from $\mathrm{TS}_{\mathrm{TTS}} = (\mathcal{S}, \mathcal{S}_0, \mathrm{Act} \cup \mathbb{R}, \Longrightarrow)$ where $(s, V) \overset{a}{\rightsquigarrow} (s', V')$ if and only if there exists a $\delta \in \mathbb{R}$ such that $(s, V) \overset{\delta}{\Longrightarrow} (s, V + \delta) \overset{a}{\Longrightarrow} (s', V')$.

Outline

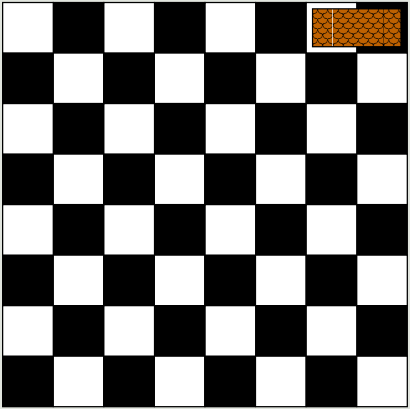
A little deviation

Tiling a chessboard...



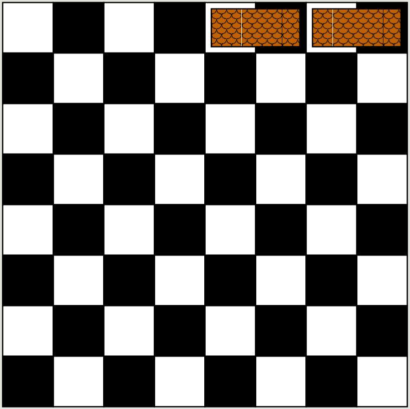
A little deviation

Tiling a chessboard...



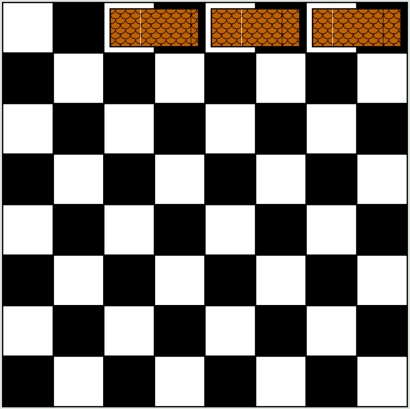
A little deviation

Tiling a chessboard...



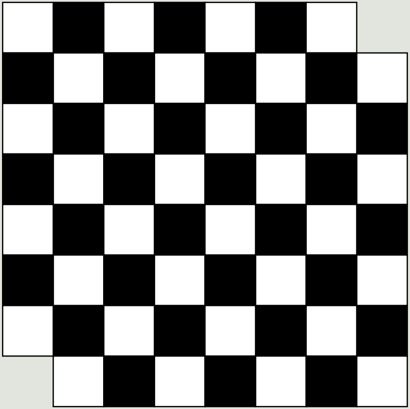
A little deviation

Tiling a chessboard...



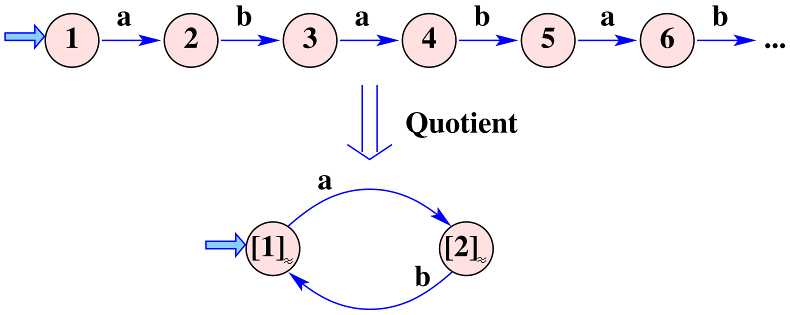
A little deviation

Tiling a chessboard missing these two squares?



Quotienting

Infinite into finite...



Quotienting

Stable equivalence relations...

Quotienting, (or partitioning by an equivalence relation[a]) is commonly used to group together objects that are similar in some sense, and hence reduce the complexity of systems. In our domain, we can use quotienting to quotient a big (infinite) transition system into a small (finite) one.

Definition: Given a transition system $\mathrm{TS} = (S, S_0, \mathrm{Act}, \Longrightarrow)$, with $\approx \subseteq S \times S$ an equivalence relation, then $\approx$ is a **stable equivalence relation** (a bisimulation) if and only if $s \approx t$ and $s \stackrel{a}{\Longrightarrow} s'$ implies that there exists t' such that $t \stackrel{a}{\Longrightarrow} t'$ and $s' \approx t'$

.

[a] An equivalence relation on a set X is a binary relation on X that is reflexive, symmetric and transitive.

Quotienting

Stable equivalence relations

$$
\begin{tikzcd}
s \arrow[r, phantom, "\approx"] \arrow[d, "a"] & t \arrow[d, "a"] \\
s' \arrow[r, phantom, "\approx"] & t'
\end{tikzcd}
$$

- A category theory diagram shows this construction.
- Since we wish to quotient infinite transition systems into finite ones, we are interested in stable equivalence relations that are finite in some sense.

Quotienting

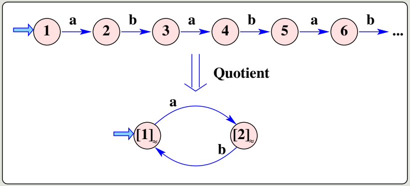
Finite stable equivalence relations

Definition: Given $\mathrm{TS} = (S, S_0, \mathrm{Act}, \Longrightarrow)$, with $\approx$ a stable equivalence relation,$[s]_\approx$ the equivalence class containing $s \in S$ (i.e. $\{s' \mid s \approx s'\}$), then $\approx$ is a **stable equivalence relation of finite index** iff $\{[s]_\approx \mid s \in S\}$ is a finite set.

Given $\mathrm{TS} = (S, S_0, \mathrm{Act}, \Longrightarrow)$, with $\approx$ a stable equivalence relation of finite index, then a new quotiented transition system is $\mathrm{QTS}_\approx = (QS, QS_0, \mathrm{Act}, \Longrightarrow)$. In this quotiented transition system, $QS = \{[s]_\approx \mid s \in S\}$ and $QS_0 = \{[s_0]_\approx \mid s_0 \in S_0\}$, and we construct $[s]_\approx \overset{a}{\Longrightarrow} [s']_\approx$ if and only if there exists $s_1 \in [s]_\approx$ and $s_1' \in [s']_\approx$ such that $s_1 \overset{a}{\Longrightarrow} s_1'$ in the transition system TS.

Quotienting

Infinite into finite



A suitable stable equivalence relation of finite index is *odd* and *even*. $i \approx j$ iff both i and j are odd, or if both i and j are even:

$$\begin{array}{lcll} \{1,3,5,\ldots\} & = & [1]_{\approx} & (= [3]_{\approx} = [5]_{\approx} = [7]_{\approx} \ldots) \\ \{2,4,6,\ldots\} & = & [2]_{\approx} & (= [4]_{\approx} = [6]_{\approx} = [6]_{\approx} \ldots) \end{array}$$