Quantitative Analysis

Quantitative Models

$x \geq 4$

$x := 0$
Quantitative Analysis

Quantitative Models

\[ x \geq 4 \]
\[ x := 0 \]

Quantitative Logics

\[ \Pr_{\leq 0.1}(\Diamond \text{error}) \]
Quantitative Models

Quantitative Logics

Quantitative Verification

$\Pr_{\leq 0.1}(\Diamond \text{error})$

$\llbracket \varphi \rrbracket(s) = 3.14$

$d(s, t) = 42$
Quantitative Analysis

Quantitative Models

Quantitative Logics

Pr_{≤1}(\Diamond \text{error})

Quantitative Verification

[\varphi](s) = 3.14

d(s, t) = 42

### Boolean world

Trace equivalence ≡

Bisimilarity ∼

s ∼ t implies s ≡ t

s ⊨ \varphi or s ⊭ \varphi

s ∼ t iff ∀\varphi : s ⊨ \varphi ⇔ t ⊨ \varphi

### “Quantification”

Linear distances \(d_L\)

Branching distances \(d_B\)

\(d_L(s, t) \leq d_B(s, t)\)

\([\varphi](s)\) is a quantity

\(d_B(s, t) = \sup_\varphi d([\varphi](s), [\varphi](t))\)
Quantitative Analysis

Problem: For processes with quantities, lots of different ways to measure distance

- point-wise
- accumulating
- limit-average
- discounting
- maximum-lead
- Cantor
- etc

\[ d_T(\sigma, \tau) = \sup_i |\sigma_i - \tau_i| \]
\[ d_T(\sigma, \tau) = \sum_i |\sigma_i - \tau_i| \]
\[ d_T(\sigma, \tau) = \lim \sup_N \frac{1}{N} \sum_{i=0}^{N} |\sigma_i - \tau_i| \]
\[ d_T(\sigma, \tau) = \sum_i \lambda^i |\sigma_i - \tau_i| \]
\[ d_T(\sigma, \tau) = \sup_N \left| \sum_{i=0}^{N} \sigma_i - \sum_{i=0}^{N} \tau_i \right| \]
\[ d_T(\sigma, \tau) = \frac{1}{1 + \inf \{j \mid \sigma_j \neq \tau_j\}} \]
Upshot

Two ideas:

- For an application, it is easiest to define distance between system traces (executions)
- Use games to convert this linear distance to branching distances

Or:

- If you give us a distance between strings, we give you back a bunch of distances between systems.
Background: Quantitative analysis

The Linear-Time–Branching-Time Spectrum via Games

From Trace Distances to Branching Distances via Games

Further Results

Conclusion
1. Background: Quantitative analysis

2. The Linear-Time–Branching-Time Spectrum via Games

3. From Trace Distances to Branching Distances via Games

4. Further Results

5. Conclusion
van Glabbeek, 2001 (excerpt):

- bisimulation eq.
- nested simulation eq.
- ready simulation eq.
- possible-futures eq.
- simulation eq.
- readiness eq.
- trace eq.
van Glabbeek, 2001 (excerpt):

- bisimulation eq. → nested simulation eq. → nested simulation pr.
- ready simulation eq. → ready simulation pr.
- ready simulation eq. → possible-futures eq. → possible-futures pr.
- simulation eq. → simulation pr.
- readiness eq. → readiness pr.
- trace eq. → trace pr.
van Glabbeek, 2001 (excerpt):

- bisimulation eq. 
- nested simulation eq. \rightarrow \text{nested simulation pr.}
- ready simulation eq. \rightarrow \text{ready simulation pr.}
- possible-futures eq. \rightarrow \text{possible-futures pr.}
- simulation eq. \rightarrow \text{simulation pr.}
- readiness eq. \rightarrow \text{readiness pr.}
- trace eq. \rightarrow \text{trace pr.}
The Simulation Game

\( s \)

\( a \)

\( b \)

\( c \)

\( t \)

\( a \)

\( b \)

\( c \)
The Simulation Game

**Spoiler**

- **a**
- **b**
- **c**

**Duplicator**

- **a**
- **b**
- **c**
The Simulation Game

**Spoiler**

- Initial state: $s$
- Transitions:
  - $s \xrightarrow{a} \bullet$
  - $\bullet \xrightarrow{b} \bullet$
  - $\bullet \xrightarrow{c} \bullet$

**Duplicator**

- Initial state: Blue circle
- Transitions:
  - Blue circle $\xrightarrow{a} \bullet$
  - $\bullet \xrightarrow{a} \bullet$
  - $\bullet \xrightarrow{b} \bullet$
  - $\bullet \xrightarrow{c} \bullet$

The Quantitative Linear-Time–Branching-Time Spectrum

Fahrenberg, Legay, Thrane
The Simulation Game

Spoiler

Duplicator

Fahrenberg, Legay, Thrane

The Quantitative Linear-Time–Branching-Time Spectrum
The Simulation Game

**Spoiler**

- **s**
  - **a**
    - **b**
    - **c**

**Duplicator**

- **t**
  - **a**
    - **b**
    - **c**
The Simulation Game

**Spoiler**

- $s$
- $a$
- $b$
- $c$

Spoiler wins

**Duplicator**

- $t$
- $a$
- $b$
- $c$

Spoiler wins
1. Player 1 ("Spoiler") chooses edge from $s$ (leading to $s'$)
2. Player 2 ("Duplicator") chooses matching edge from $t$ (leading to $t'$)
3. Game continues from configuration $s'$, $t'$
\[ \omega \] If Player 2 can always answer: YES, $t$ simulates $s$. Otherwise: NO
The Linear-Time–Branching-Time Spectrum, Reordered

bisimulation eq.

3-nested simulation pr.

2-nested simulation eq.

2-nested ready sim. eq.

2-nested ready sim. pr.

2-nested simulation pr.

ready simulation eq.

simulation eq.

ready simulation pr.

simulation pr.
1. Background: Quantitative analysis

2. The Linear-Time–Branching-Time Spectrum via Games

3. From Trace Distances to Branching Distances via Games

4. Further Results

5. Conclusion
The Simulation Game, Revisited

1. Player 1 chooses edge from $s$ (leading to $s'$)
2. Player 2 chooses matching edge from $t$ (leading to $t'$)
3. Game continues from configuration $s', t'$

$\omega$. If Player 2 can always answer: YES, $t$ simulates $s$.
Otherwise: NO

Or, as an Ehrenfeucht-Fraïssé game:

1. Player 1 chooses edge from $s$ (leading to $s'$)
2. Player 2 chooses edge from $t$ (leading to $t'$)
3. Game continues from new configuration $s', t'$

$\omega$. At the end (maybe after infinitely many rounds!), compare the chosen traces:
If the trace chosen by $t$ matches the one chosen by $s$: YES
Otherwise: NO
Quantitative Ehrenfeucht-Fraïssé Games

The quantitative setting:

- Assume we have a way, possibly application-determined, to measure distances of (finite or infinite) traces
- Hence a (hemi)metric $d_T : (\sigma, \tau) \mapsto d_T(\sigma, \tau) \in \mathbb{R}_{\geq 0} \cup \{\infty\}$

The quantitative Ehrenfeucht-Fraïssé game:

1. Player 1 chooses edge from $s$ (leading to $s'$)
2. Player 2 chooses edge from $t$ (leading to $t'$)
3. Game continues from new configuration $s', t'$

$\omega$. At the end, compare the chosen traces $\sigma, \tau$:
- The simulation distance from $s$ to $t$ is defined to be $d_T(\sigma, \tau)$

This can be done for all the games in the LTBT spectrum.
Quantitative EF Games: The Gory Details – 1

- **Configuration** of the game: \((\pi, \rho)\): \(\pi\) the Player-1 choices up to now; \(\rho\) the Player-2 choices
- **Strategy**: mapping from configurations to next moves
  - \(\Theta_i\): set of Player-\(i\) strategies
- **Simulation strategy**: Player-1 moves allowed from end of \(\pi\)
- **Bisimulation strategy**: Player-1 moves allowed from end of \(\pi\) or end of \(\rho\)
  - (hence \(\pi\) and \(\rho\) are generally not paths – “mingled paths”)
- Pair of strategies \(\Rightarrow\) (possibly infinite) sequence of configurations
- Take the limit; unmingle \(\Rightarrow\) pair of (possibly infinite) traces \((\sigma, \tau)\)
- **Bisimulation distance**: \(\sup_{\theta_1 \in \Theta_1} \inf_{\theta_2 \in \Theta_2} d_T(\sigma, \tau)\)
- **Simulation distance**: \(\sup_{\theta_1 \in \Theta_1^0} \inf_{\theta_2 \in \Theta_2} d_T(\sigma, \tau)\) (restricting Player 1’s capabilities)
Blind Player-1 strategies: depend only on the end of $\rho$
- (“cannot see Player-2 moves”)
- $\tilde{\Theta}_1$: set of blind Player-1 strategies

Trace inclusion distance: $\sup_{\theta_1 \in \tilde{\Theta}_1^0} \inf_{\theta_2 \in \Theta_2} d_T(\sigma, \tau)$

For nesting: count the number of times Player 1 chooses edge from end of $\rho$
- $\Theta_1^k$: $k$ choices from end of $\rho$ allowed

Nested simulation distance: $\sup_{\theta_1 \in \Theta_1^1} \inf_{\theta_2 \in \Theta_2} d_T(\sigma, \tau)$

Nested trace inclusion distance: $\sup_{\theta_1 \in \tilde{\Theta}_1^1} \inf_{\theta_2 \in \Theta_2} d_T(\sigma, \tau)$

For ready: allow extra “I’ll see you” Player-1 transition from end of $\rho$
For any trace distance $d : (\sigma, \tau) \mapsto d(\sigma, \tau) \in \mathbb{R}_{\geq 0} \cup \{\infty\}$:
1. Background: Quantitative analysis

2. The Linear-Time–Branching-Time Spectrum via Games

3. From Trace Distances to Branching Distances via Games

4. Further Results

5. Conclusion
Given two equivalences or preorders in the *qualitative* setting for which there is a *counter-example* which separates them, then the two corresponding distances are *topologically inequivalent*

(under certain mild conditions for the trace distance)

(And the proof uses precisely the same counter-example!)

---

Fahrenberg, Legay, Thrane

The Quantitative Linear-Time–Branching-Time Spectrum
Recursive Characterization

- If the trace distance $d : (\sigma, \tau) \mapsto d(\sigma, \tau)$ has a decomposition $d = g \circ f : \text{Tr} \times \text{Tr} \to L \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ through a complete lattice $L$,
- and $f$ has a recursive formula
- i.e. such that $f(\sigma, \tau) = F(\sigma_0, \tau_0, f(\sigma^1, \tau^1))$ for some $F : \Sigma \times \Sigma \times L \to L$ (which is *monotone* in the third coordinate)
- (where $\sigma = \sigma_0 \cdot \sigma^1$ is a split of $\sigma$ into first element and tail)
- then all distances in the QLTBT are given as least fixed points of some functionals using $F$

All trace distances we know can be expressed recursively like this.
Recursive Characterization: Theorem

The endofunction $I$ on $(\mathbb{N}_+ \cup \{\infty\}) \times \{1, 2\} \to L^{S \times S}$ defined by

$$I(h_{m,p})(s, t) = \begin{cases} \max \left\{ \sup_{s \xrightarrow{x} s'} \inf_{t \xrightarrow{y} t'} F(x, y, h_{m,1}(s', t')) \right\} & \text{if } m \geq 2, p = 1 \\ \max \left\{ \sup_{t \xrightarrow{y} t'} \inf_{s \xrightarrow{x} s'} F(x, y, h_{m-1,2}(s', t')) \right\} & \text{if } m = 1, p = 1 \\ \sup_{s \xrightarrow{x} s'} \inf_{t \xrightarrow{y} t'} F(x, y, h_{m,1}(s', t')) & \text{if } m = 1, p = 2 \\ \sup_{t \xrightarrow{y} t'} \inf_{s \xrightarrow{x} s'} F(x, y, h_{m-1,1}(s', t')) & \text{if } m \geq 2, p = 2 \\ \sup_{t \xrightarrow{y} t'} \inf_{s t \xrightarrow{x} s'} F(x, y, h_{m,2}(s', t')) & \text{if } m = 1, p = 2 \\ \sup_{t \xrightarrow{y} t'} \inf_{s t \xrightarrow{x} s'} F(x, y, h_{m,2}(s', t')) & \end{cases}$$

has a least fixed point $h^* : (\mathbb{N}_+ \cup \{\infty\}) \times \{1, 2\} \to L^{S \times S}$, and if the LTS $(S, T)$ is finitely branching, then $d'^{k\sim} = g \circ h^*_{k,1}$ for all $k \in \mathbb{N}_+ \cup \{\infty\}$. 
Conclusion & Further Work

- We show how to convert any (typically application-given) distance on system traces to (almost) any type of branching distance in the LTBT spectrum
- “Adding an extra dimension to the LTBT spectrum”

- Application to different scenarios (How does it work in concrete cases? Do we get sensible algorithms? Approximations?)
- Application to real-time and hybrid systems
  - Replace “finitely branching” by “compactly branching”?
- Quantitative LTBT with silent moves?
- What about probabilistic systems?