• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• The training logs contain no noise
  – All traces in the training logs are to be classified as positive
• The winner is the one that classifies the most traces correctly
  – The readability of the model is not relevant
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• A collection of discovered accepting Petri nets
  – Initial marking
  – Set of final markings
• Semantics:
  – Replay the trace-to-classify on every discovered net and accumulate the replay costs
    • Costs for move-on-log: 10
    • Costs for move-on-model: 4
    • Other costs: 0
  – Classify the trace-to-classify as positive if and only if the accumulated replay costs are 0
  – Use decomposed replay to speed up the replay
    • Decomposed replay preserves perfect fitness, that is, costs 0
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• Use decomposed discovery
  – Preserves perfect fitness
• Use only discovery algorithms that guarantee perfect fitness
  – ILP Miner
  – Hybrid ILP Miner
  – Inductive Miner (variant that guarantees perfect fitness)
  – …?

• As a result, all traces of the training logs will be classified as positive
• Use as many decomposed discovery algorithms as are useful
  – An algorithm is considered to be useful if adding it results in additional negatives

• Note that we already have guaranteed that all traces from the training logs will be classified as positive. With adding additional discovery algorithms, we try to squeeze out as many negatives as possible.
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• We used the April and May logs to select the best set of useful decomposed discovery algorithms:
  – As few algorithms as needed to achieve the best result
• The best result (showing numbers of traces classified as negative):

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training log</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>April log</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>May log</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

• As an example, we classified in the May 3 log only 8 traces as negatives. Hence, we know that we have at least 2 false positives.
• The best set of useful decomposed discovery algorithms:
  – Non-decomposed Hybrid ILP Miner (HIM-0)
  – Maximal-decomposed Inductive Miner (IM-100)
Result from the April and May logs, as confirmed by organizers:
- 1 misclassification for the April logs
- 5 misclassifications for the May logs
- These numbers match the numbers of ‘known’ false positives

<table>
<thead>
<tr>
<th>Training log</th>
<th>F</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>April log</td>
<td></td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>May log</td>
<td></td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

- No false negatives!
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• The DrFurby Classifier enriches the log-to-classify, using a “drfurby” extension:
  – Log attributes:
    • name: Name of the log-to-classify
    • positive: Number of traces classified as positive
    • negative: Number of traces classified as negative
    • millis: Number of milliseconds it took to classify the log
  – Trace attributes:
    • classification: “positive” or “negative”
    • him0Costs: Costs of replaying this trace on the net as discovered by IM-100
    • im100Costs: Costs of replaying this trace on the net as discovered by HIM-0
    • totalCosts: Accumulated costs of replaying this trace on all discovered nets
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
Implementation: DrFurby Classifier plug-in in ProM 6.6 (DivideAndConquer package)

The inputs

The plug-in

The outputs
Implementation: result of HIM0 on log 3
Implementation: result of IM100 on log 3
Implementation: result of IM100 on log 3
Implementation: result of IM100 on log 3
• Observations
• Model
• Discovery
• Algorithm selection
• Extension
• Implementation
• Contest
• Conclusions
• Results on the June logs:
  – Results for April and May logs added to allow comparison

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training log</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>April log</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>May log</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>June log</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

• 7 misclassifications
  – 7 false positives
  – No false negatives
Overview

- Observations
- Model
- Discovery
- Algorithm selection
- Extension
- Implementation
- Contest
- Conclusions
• DrFurby Classifier
  – Takes a training log and a test log
    • Assumes training log to be free of noise
  – Classifies the traces in the test log using the training log
• Some problems with some logs
  – Like 3 and 8
• No false negatives, only false positives
  – No guarantee on the former, however
Questions?
• Backup
• “DrFurby”
  – Is pronounced almost identically as “dr. Verbeek” 😊
  – Happened years ago, Boudewijn knows the story, he was there.
• Why decomposition?
  – Results on the April and May logs:

<table>
<thead>
<tr>
<th>Training log</th>
<th>F</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>April log</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>May log</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>June log</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

  – Results on the same logs without using decomposition:

<table>
<thead>
<tr>
<th>Training log</th>
<th>F</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>April log</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>May log</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>June log</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

  – No change for 9 out of 10 logs, but a significant improvement for log 3