Systematic Simplicity-Accuracy Tradeoffs in Parameterized Contract Models

Ian D. Peake    Heinz W. Schmidt

School of Computer Science and Information Technology, RMIT University, Melbourne, Australia

QoSA 2011, Boulder, Colorado
Radiation dose at Fukushima Daiichi Nuclear Power Station, The Main Building

http://api.pachube.com/v2/feeds/24856.json


Feed the data retrieved from http://www.tepco.co.jp/. This...
Pachube Plans (Parameterized Contract)

**Pachube Basic**
- Public feeds
- 1 month historical storage
- 5 datastreams
- 5 API requests / minute
- Import 500 existing datapoints / day

**FREE**

**Pachube Pro**

**Pachube Recommended**
- Public feeds
- 1 year historical storage
- 40 datastreams
- 40 API requests / minute
- Import 4,000 existing datapoints / day

$1.99 / month on an annual plan

**Choose Plan**

or $2.99 / month

**Pachube Premium**

- **Private and public feeds**
- **Unlimited** historical storage
- 250 datastreams
- 250 API requests / minute
- Import unlimited existing datapoints

$7.99 / month on an annual plan

**Choose Plan**

or $11.99 / month

We offer customized plans with higher bandwidth and a provisioning server for commercial deployments. Contact us for more info.
What is suitable *generic* model for parameterized contracts?

Propose

- *Feature Dependence Relations (FDRs)*
- *Galois parameterized Contracts (GCs)* generalizing FDRs

GCs compatible with existing notations:

- Tables
- Trace automata

Enable accuracy-simplicity tradeoffs:

- between GCs
- between GC-compatible notations
### Production Cell - Cycle Time - Contract Table

- Production Cell [Lewerentz95]
- Modeling Predictable Component Architectures [Schmidt03]
- Contract (table)-based WCET [Fredriksson07]

<table>
<thead>
<tr>
<th></th>
<th>max(CT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>$t_1$</td>
</tr>
<tr>
<td>safe</td>
<td>$t_2$</td>
</tr>
<tr>
<td>1arm</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>$t_{max}$</td>
</tr>
</tbody>
</table>
Feature Dependence Relation (FDR)

\[
\begin{array}{|c|c|}
\hline
\text{max(CT)} & \text{t} \_1 \\
\hline
\text{fast} & \text{t} \_1 \\
\hline
\text{safe} & \text{t} \_2 \\
\hline
\ldots & \text{t} \_\text{max} \\
\hline
\end{array}
\]

\[
\downarrow
\]

\[
\begin{array}{|c|c|c|c|}
\hline
\text{P} & \text{R} & \text{t} \_1 & \text{t} \_2 & \text{t} \_\text{max} \\
\hline
\text{fast} & \text{X} & \text{X} & \text{X} \\
\hline
\text{safe} & \text{X} & \text{X} & \text{X} \\
\hline
\ldots & \text{X} & \text{X} & \text{X} \\
\hline
\end{array}
\]
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Galois connection
Lattices

Peake, Schmidt (RMIT University)  Simplicity-Accuracy Tradeoffs  17th June 2011  7 / 13
### Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Galois connection

Lattices

Peake, Schmidt (RMIT University) Simplicity-Accuracy Tradeoffs

17th June 2011 7 / 13
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection

Lattices

Peake, Schmidt (RMIT University)  Simplicity-Accuracy Tradeoffs  17th June 2011
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection

Lattices

Peake, Schmidt (RMIT University)
Simplicity-Accuracy Tradeoffs
17th June 2011
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\[
\{ \text{fast} \} \rightarrow \{ \text{safe} \} \rightarrow \{ \text{fast} \} \rightarrow \{ t_1 \}
\]

\[
\{ \text{safe, } \ldots \} \rightarrow \{ \text{fast, } \ldots \} \rightarrow \{ \text{fast, safe} \} \rightarrow \{ t_1, t_2 \}
\]

\[
\{ \text{fast, safe, } \ldots \} \rightarrow \{ t_1, t_2, t_{\text{max}} \}
\]
### Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

---

#### Lattices

- Galois connection
- Simplicity-Accuracy Tradeoffs

---

**Notes:**

- Peake, Schmidt (RMIT University)
- 17th June 2011
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The diagram illustrates the feature dependence relation, where X represents an assertion of dependence between features or time points. The arrows indicate the direction of dependence, with dotted lines possibly indicating indirect or complex relationships. The labels on the nodes are sets of features or time points, reflecting the simplicities and accuracies tradeoffs in the production cell context.
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection + Lattices
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection  +  Lattices = Galois contract (GC)
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection + Lattices = Galois contract (GC)
Production Cell - Feature Dependence Relation

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safe</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Galois connection + Lattices = Galois contract (GC)
Simplicity vs Accuracy: between GCs

**Generalization:** Say $c'$ generalizes $c$ if and only if, all else equal, $c'$ requires no less and provides no more than $c'$. 
**Generalization:** Say \( c' \) generalizes \( c \) if and only if, all else equal, \( c' \) requires no less and provides no more than \( c' \).

Here \( c' \) is simpler, less accurate than \( c \).
Simplicity vs Accuracy: between GCs

**Generalization:** Say \( c' \) generalizes \( c \) if and only if, all else equal, \( c' \) requires no less and provides no more than \( c' \).

Here \( c' \) simpler, less accurate than \( c \).
Automata as FDRs

Can model automata properties using FDRs, GCs.

(E.g.) Interpret regular (or trace) language as FDR via *(trace language) translation relation* [Schmidt02]. Interpret traces (possible behaviour or behaviour aspects) as features.
Tradeoffs between GC-compatible Notations

Each compatible notation has own (“local”) tradeoffs. E.g.:

- Tables simple, intuitive, not suited to large systems.
- Automata more accurate for large systems, more complex.
Each compatible notation has own ("local") tradeoffs. E.g.:
- Tables simple, intuitive, not suited to large systems.
- Automata more accurate for large systems, more complex.

GCs may help unify contracts at syntactic, behavioural, synchronisation and extra-functional levels.
Tradeoffs between GC-compatible Notations

Each compatible notation has own (“local”) tradeoffs. E.g.:
- Tables simple, intuitive, not suited to large systems.
- Automata more accurate for large systems, more complex.

GCs may help unify contracts at syntactic, behavioural, synchronisation and extra-functional levels.

GCs generalizable via any compatible notation (at least trivially).
Tradeoffs between GC-compatible Notations

Each compatible notation has own ("local") tradeoffs. E.g.:
- Tables simple, intuitive, not suited to large systems.
- Automata more accurate for large systems, more complex.

GCs may help unify contracts at syntactic, behavioural, synchronisation and extra-functional levels.

GCs generalizable via any compatible notation (at least trivially).

(Hypothesise:) Arbitrary GCs modelable in any compatible notation to useful accuracy.
Conclusion

Propose Feature Dependence Relations (FDRs) and Galois parameterized Contracts (GCs)

- Systematic accuracy-simplicity tradeoffs
- Compatible with existing models (tables, automata)

Future:

- Connect to abstract interpretation, logic, assume-guarantee
- Retrofit to our RADL ADL, bridge to other tools e.g. PRISM

Vision: Simple generic method: Given logic, model \( m \), arbitrary (prov-/req-) formulae. Derive closest FDR generalizing \( m \).


GC vs explicit GC model:

\[
\begin{align*}
\{\} & \quad \text{undefined} \\
\{\ldots\}, \{\text{safe}\}, \{\text{fast}\} & \quad t_1 \\
\{\text{fast, ...}\}, \{\text{fast, safe}\} & \quad t_2 \\
\{\text{fast, safe, ...}\} & \quad t_{\text{Max}}
\end{align*}
\]
Automata as FDRs [Mazurkiewicz95]

Trace: partially-ordered event set.

Trace language: set of traces.

Projection ($\pi \Sigma (t)$): Given trace $t$ and set $\Sigma$, keep only events in $\Sigma$.

E.g.: $\pi \text{press}^\ast (\text{safe.press}) = \text{press}$

Translation Relation [Schmidt02]: Given sets $I, O$; traces $t \in T$, relate $\pi I (t)$ to $\pi O (t)$.
Automata as FDRs [Mazurkiewicz95]

**Trace**: partially-ordered event set.
Automata as FDRs [Mazurkiewicz95]

**Trace**: partially-ordered event set. **Trace language**: set of traces.

fast.press:

```
<table>
<thead>
<tr>
<th>a1.X(1)</th>
<th>a1.take</th>
<th>tl(1)</th>
<th>a1.R(1)</th>
<th>tl(2)</th>
<th>a1.drop</th>
<th>a1.R(2)</th>
<th>press</th>
</tr>
</thead>
<tbody>
<tr>
<td>press.down</td>
<td>a2.X(1)</td>
<td>a2.take</td>
<td>a2.R(1)</td>
<td>press.up</td>
<td>a2.X(2)</td>
<td>a2.drop</td>
<td>a2.R(2)</td>
</tr>
<tr>
<td>tr(1)</td>
<td>tr(2)</td>
<td>tr(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Projection ($\pi$): Given trace $t$ and set $\Sigma$, keep only events in $\Sigma$.

E.g.: $\pi$ press $\ast$ (safe.press) =

Translation Relation [Schmidt02]: Given sets $I, O$; traces $t \in T$, relate $\pi_I(t)$ to $\pi_O(t)$.

Peake, Schmidt (RMIT University) Simplicity-Accuracy Tradeoffs 17th June 2011 13 / 13
Automata as FDRs [Mazurkiewicz95]

**Trace**: partially-ordered event set. **Trace language**: set of traces.

**fast.press:**

1. **a1.X(1)**
2. **a1.take**
3. **t(1)**
4. **a2.X(1)**
5. **a2.take**
6. **t(2)**
7. **a2.R(1)**
8. **press.up**
9. **a2.X(2)**
10. **a1.drop**
11. **a1.R(2)**
12. **press**
13. **a2.drop**
14. **tr(1)**
15. **tr(2)**
16. **tr(3)**

**safe.press:**

1. **a1.X(1)**
2. **a1.take**
3. **t(1)**
4. **a2.X(1)**
5. **a2.take**
6. **t(2)**
7. **a2.R(1)**
8. **press.up**
9. **a1.X(2)**
10. **a1.drop**
11. **a1.R(2)**
12. **press**
13. **a2.X(2)**
14. **a2.drop**
15. **tr(1)**
16. **tr(2)**
17. **tr(3)**
**Automata as FDRs [Mazurkiewicz95]**

**Trace**: partially-ordered event set.  
**Trace language**: set of traces.

**fast.press:**

```
1.X(1) → a1.take → a1.R(1) → tl(1) → a2.take → a2.R(1) → press.up → tl(2) → a1.drop → a1.R(2) → press
```

**safe.press:**

```
1.X(1) → a1.take → a1.R(1) → tl(1) → a2.take → a2.R(1) → press.up → tl(2) → a1.drop → a1.R(2) → press
```

**Projection ($\pi_\Sigma(t)$):** Given trace $t$ and set $\Sigma$, keep only events in $\Sigma$.

---

Peake, Schmidt (RMIT University)  
Simplicity-Accuracy Tradeoffs  
17th June 2011
Automata as FDRs [Mazurkiewicz95]

**Trace**: partially-ordered event set.  
**Trace language**: set of traces.

**fast.press:**

```
(a1.X(1) -> a1.take) -> tl(1) -> a1.R(1) -> a2.take -> tl(2) -> a2.R(1) -> press.up -> tl(3) -> a2.drop -> a2.R(2) -> press 
press.down -> a2.X(1) -> press
```

**safe.press:**

```
(a1.X(1) -> a1.take) -> tl(1) -> a1.R(1) -> a2.take -> tl(2) -> a2.R(1) -> press.up -> tl(3) -> a2.drop -> a2.R(2) -> tr(1) -> tr(2) -> tr(3) 
press.down -> a2.X(1) -> press
```

**Projection** ($\pi_\Sigma(t)$): Given trace $t$ and set $\Sigma$, keep only events in $\Sigma$.

E.g.: $\pi_{press^*}(safe.press) = press.down -> press.up -> press$
Automata as FDRs [Mazurkiewicz95]

**Trace**: partially-ordered event set.  **Trace language**: set of traces.

**fast.press:**

```
<table>
<thead>
<tr>
<th>a1.X(1)</th>
<th>a1.take</th>
<th>a1.R(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>press.down</td>
<td>a2.X(1)</td>
<td>a2.R(1)</td>
</tr>
<tr>
<td></td>
<td>t(1)</td>
<td>press.up</td>
</tr>
<tr>
<td></td>
<td>a2.take</td>
<td>a2.X(2)</td>
</tr>
<tr>
<td></td>
<td>t(l2)</td>
<td>a1.drop</td>
</tr>
<tr>
<td></td>
<td>a1.X(2)</td>
<td>a1.R(2)</td>
</tr>
<tr>
<td></td>
<td>press</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a2.drop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(3)</td>
<td></td>
</tr>
</tbody>
</table>
```

**safe.press:**

```
<table>
<thead>
<tr>
<th>a1.X(1)</th>
<th>a1.take</th>
<th>a1.R(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>press.down</td>
<td>a2.X(1)</td>
<td>a2.R(1)</td>
</tr>
<tr>
<td></td>
<td>t(l1)</td>
<td>press.up</td>
</tr>
<tr>
<td></td>
<td>a2.take</td>
<td>a2.X(2)</td>
</tr>
<tr>
<td></td>
<td>t(l2)</td>
<td>a1.drop</td>
</tr>
<tr>
<td></td>
<td>a1.X(2)</td>
<td>a1.R(2)</td>
</tr>
<tr>
<td></td>
<td>press</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a2.drop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tr(3)</td>
<td></td>
</tr>
</tbody>
</table>
```

**Projection** ($\pi_{\Sigma}(t)$): Given trace $t$ and set $\Sigma$, keep only events in $\Sigma$.

E.g.: $\pi_{\text{press}^*}(\text{safe.press}) = \text{press.down} \rightarrow \text{press.up} \rightarrow \text{press}$

**Translation Relation** [Schmidt02]:

Given sets $I,O$; traces $t \in T$, relate $\pi_I(t)$ to $\pi_O(t)$. 