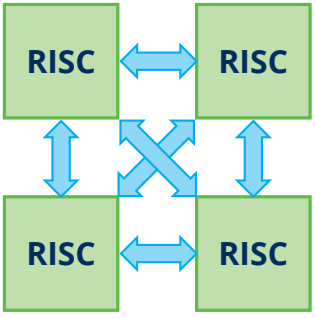
ProFeat: Quantitative Analysis of Feature-oriented Systems

Philipp Chrszon

2019-03-14

Chair for Algebraic and Logical Foundations of Computer Science
Institute of Theoretical Computer Science – TU Dresden

Case study: heterogeneous tiled architecture

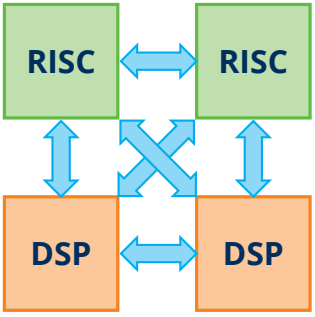


Architecture

- processing units (tiles) can be mixed freely
- connected via network-on-chip

Christel Baier, Sascha Klüppelholz, Sascha Wunderlich
Towards Automated Variant Selection for Heterogenous Tiled Architectures, Models, Algorithms, Logics and Tools, LNCS 10460, pp 382–399, 2017.

Case study: heterogeneous tiled architecture

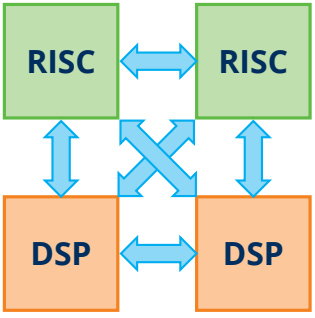


Architecture

- processing units (tiles) can be mixed freely
- connected via network-on-chip

Christel Baier, Sascha Klüppelholz, Sascha Wunderlich
Towards Automated Variant Selection for Heterogenous Tiled Architectures, Models, Algorithms, Logics and Tools, LNCS 10460, pp 382–399, 2017.

Case study: heterogeneous tiled architecture



Architecture

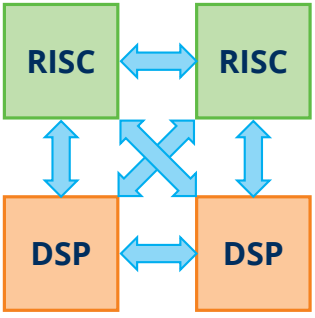
- processing units (tiles) can be mixed freely
- connected via network-on-chip

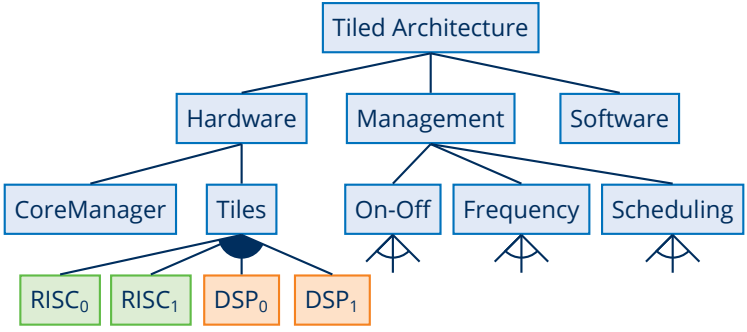
Variant space

- combination of tiles
- strategies for power saving, frequency scaling, scheduling

Christel Baier, Sascha Klüppelholz, Sascha Wunderlich
Towards Automated Variant Selection for Heterogenous Tiled Architectures, Models, Algorithms, Logics and Tools, LNCS 10460, pp 382–399, 2017.

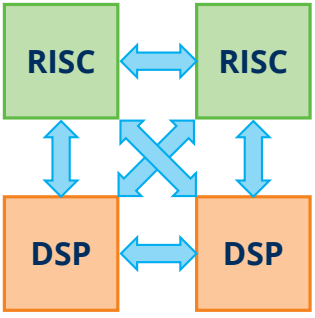
Case study: heterogeneous tiled architecture

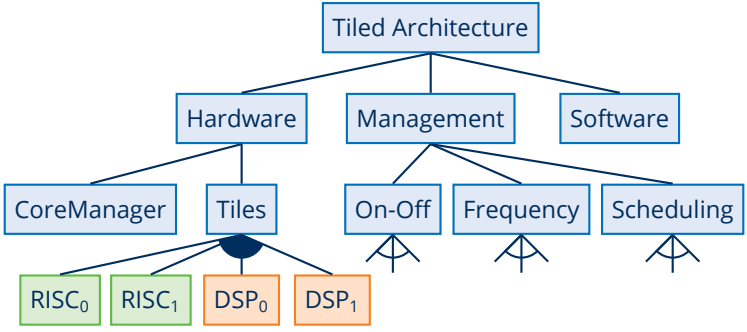




Christel Baier, Sascha Klüppelholz, Sascha Wunderlich
Towards Automated Variant Selection for Heterogenous Tiled Architectures, Models, Algorithms, Logics and Tools, LNCS 10460, pp 382–399, 2017.

Case study: heterogeneous tiled architecture

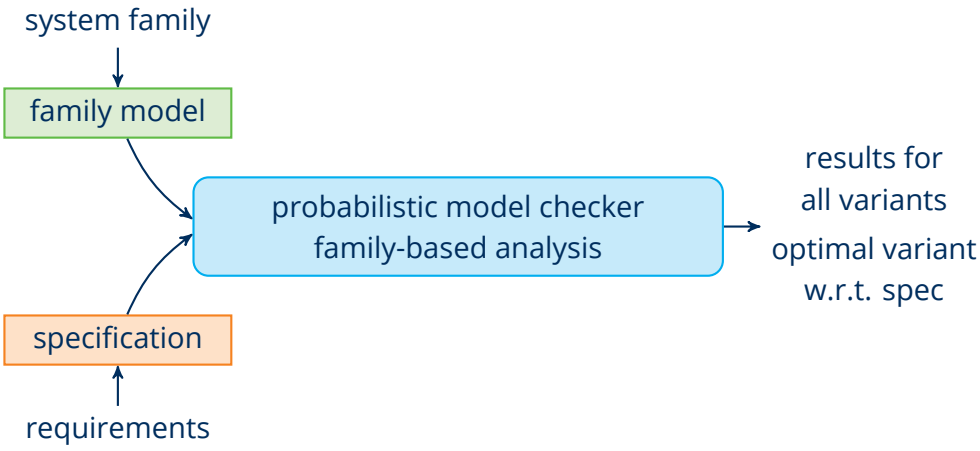




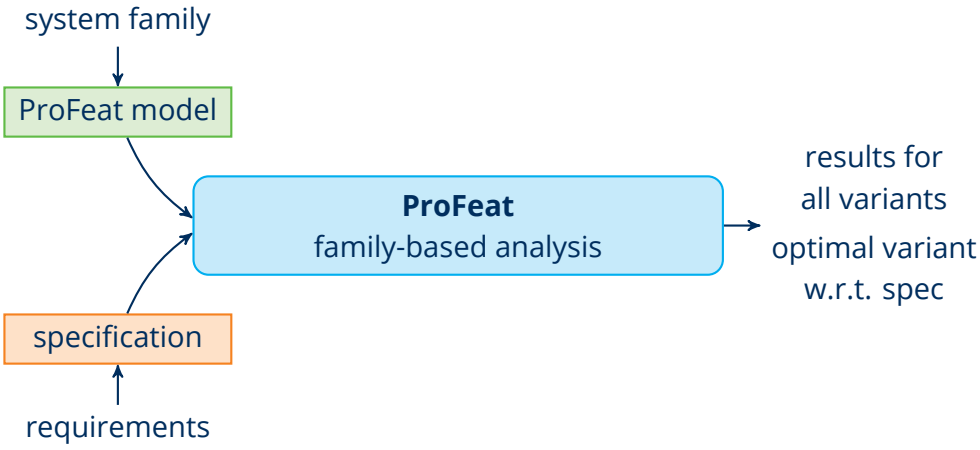
Goals

- compare homogeneous and heterogeneous systems w.r.t. energy consumption and throughput
- analysis of resource management strategies

Approach: probabilistic model checking



Approach: probabilistic model checking

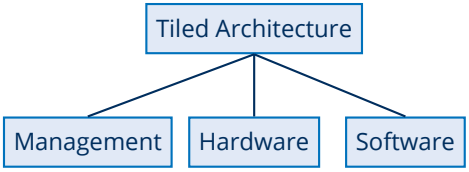


Philipp Chrszon, Clemens Dubslaff, Sascha Klüppelholz, Christel Baier.
ProFeat: Feature-oriented Engineering for Family-based Probabilistic Model Checking.
Formal Aspects of Computing 30(1), pp 45–75, 2018.

Feature modeling

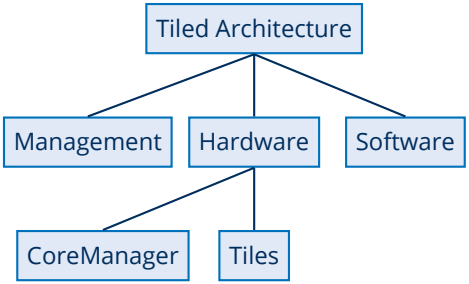
Feature modeling

```
root feature {
   all of Hardware, Software, Management;
}
```



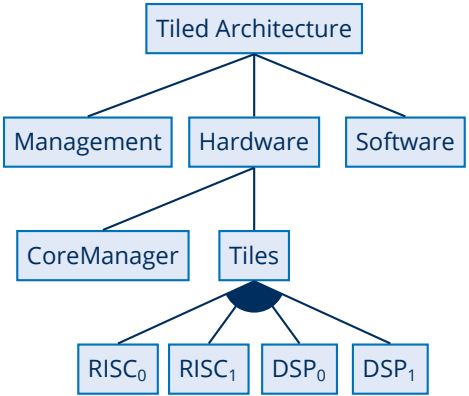
Feature modeling

```
root feature {
   all of Hardware, Software, Management;
}

feature Hardware {
   all of CoreManager, Tiles;
}
```



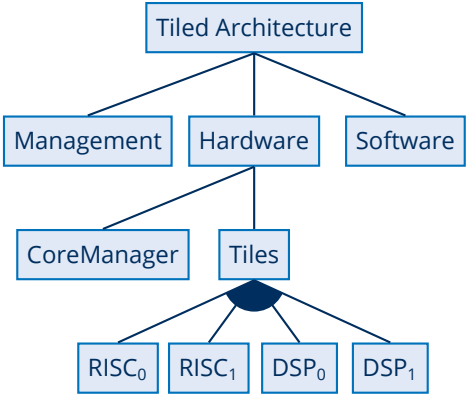
Feature modeling

```
root feature {
   all of Hardware, Software, Management;
}

feature Hardware {
   all of CoreManager, Tiles;
}

feature Tiles {
   some of RISC[2], DSP[2];
}
```



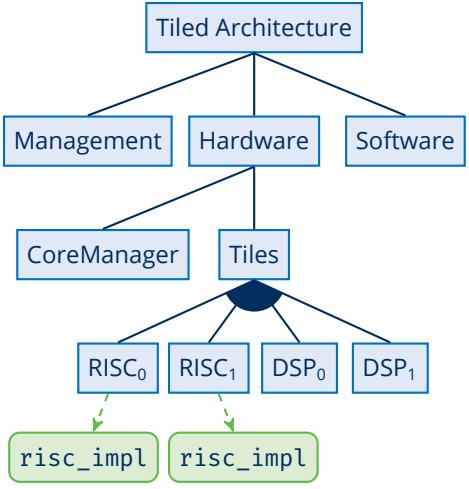
Feature modeling

```
root feature {
   all of Hardware, Software, Management;
}

feature Hardware {
   all of CoreManager, Tiles;
}

feature Tiles {
   some of RISC[2], DSP[2];
}

feature RISC {
   frequency : [0..MAX_FREQ];
```



Feature modeling

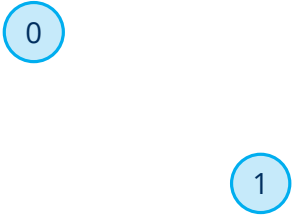
```
root feature {
   all of Hardware, Software, Management;
}

feature Hardware {
   all of CoreManager, Tiles;
}

feature Tiles {
   some of RISC[2], DSP[2];
}

feature RISC {
   frequency : [0..MAX_FREQ];
   modules risc_impl;
}
```



Feature modules: defining operational behavior

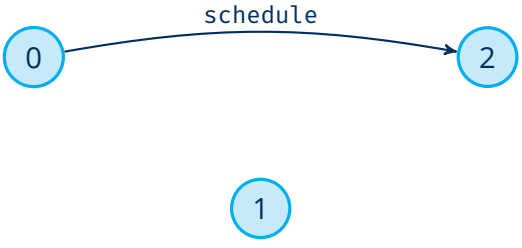
Feature modules: defining operational behavior

```
module RISC {
  tasks : [0..MAX_TASKS] init 0;




}
```



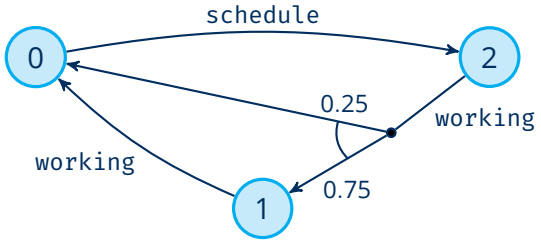
Feature modules: defining operational behavior

```
module RISC {
  tasks : [0..MAX_TASKS] init 0;

  [schedule] tasks = 0 -> (tasks' = 2);



}
```



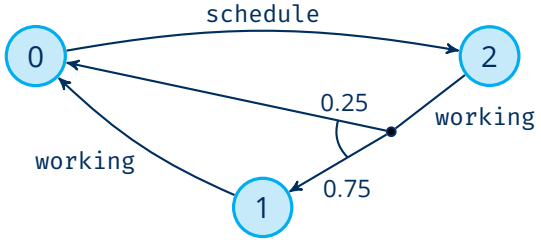
Feature modules: defining operational behavior

```
module RISC {
  tasks : [0..MAX_TASKS] init 0;

  [schedule] tasks = 0 -> (tasks' = 2);
  [working] tasks > 0 -> 0.75: (tasks' = max(0, tasks - 1)) +
                         0.25: (tasks' = max(0, tasks - 2));

}
```



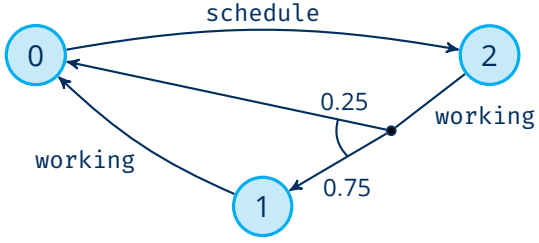
Feature modules: defining operational behavior

```
module RISC {
  tasks : [0..MAX_TASKS] init 0;

  [schedule] tasks = 0 -> (tasks' = 2);
  [working] tasks > 0 -> 0.75: (tasks' = max(0, tasks - frequency)) +
                         0.25: (tasks' = max(0, tasks - 2 * frequency));

}
```



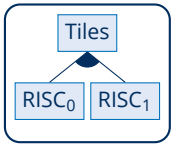
Feature modules: defining operational behavior

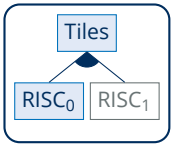
```
module RISC {
  tasks : [0..MAX_TASKS] init 0;

  [schedule] tasks = 0 -> (tasks' = 2);
  [working] tasks > 0 -> 0.75: (tasks' = max(0, tasks - frequency)) +
                         0.25: (tasks' = max(0, tasks - 2 * frequency));
  [deactivate] tasks = 0 -> true;
}
```

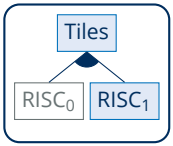


Feature controller: switching between configurations

```
controller {




}
```

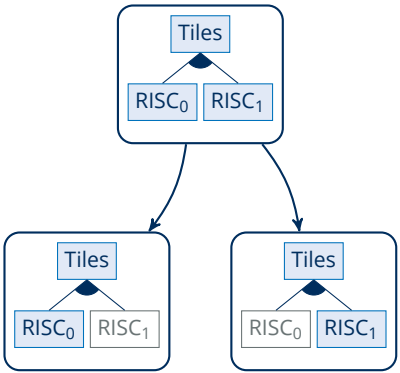






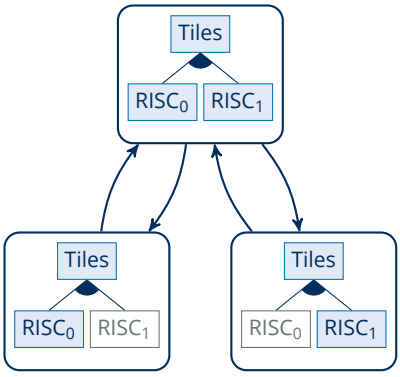
Feature controller: switching between configurations

```
controller {
  for i in [0..1] {
    [] active(Management.OnOff.Powersave)
       & can_power_off(RISC[i]) ->
         deactivate(RISC[i]);
  }
}
```



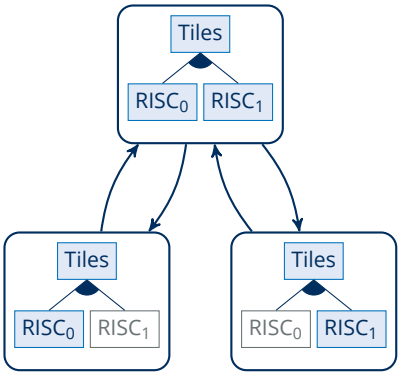
Feature controller: switching between configurations

```
controller {
  for i in [0..1] {
    [] active(Management.OnOff.Powersave)
       & can_power_off(RISC[i]) ->
         deactivate(RISC[i]);
  }
}
```



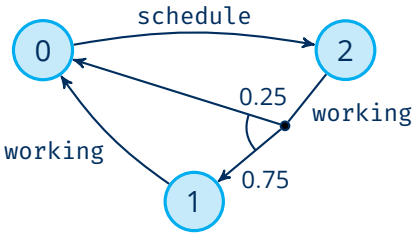
Feature controller: switching between configurations

```
controller {
  for i in [0..1] {
    [] active(Management.OnOff.Powersave)
       & can_power_off(RISC[i]) ->
         deactivate(RISC[i]);

    [] active(Management.Frequency.NonDet)
      -> (RISC[i].frequency' = 1);
    [] active(Management.Frequency.NonDet)
      -> (RISC[i].frequency' = 2);
  }
}
```



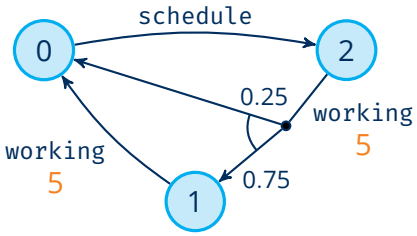
Costs and rewards

```
feature RISC {
  frequency : [0..MAX_FREQ];
  modules risc_impl;




}
```



Costs and rewards

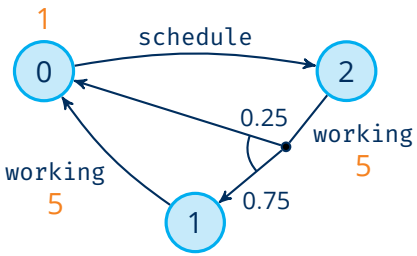
```
feature RISC {
  frequency : [0..MAX_FREQ];
  modules risc_impl;

  rewards "energy" {
    [working] true :
      1 + pow(frequency, 2);


  }
}
```



variant with `frequency=2`

Costs and rewards

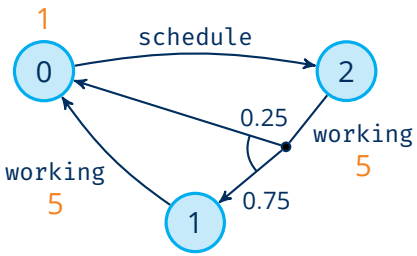
```
feature RISC {
  frequency : [0..MAX_FREQ];
  modules risc_impl;

  rewards "energy" {
    [working] true :
      1 + pow(frequency, 2);
    active(this) & this.tasks = 0 : 1;

  }
}
```



variant with `frequency=2`

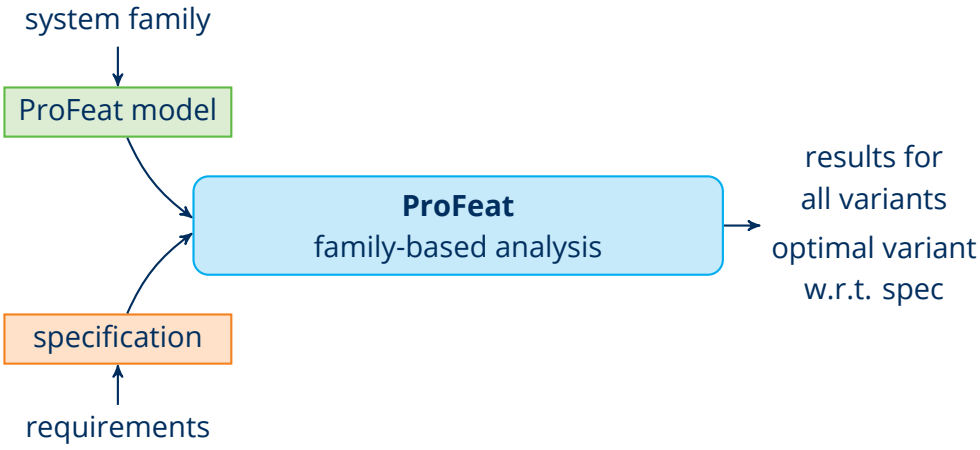
Costs and rewards

```
feature RISC {
  frequency : [0..MAX_FREQ];
  modules risc_impl;

  rewards "energy" {
    [working] true :
      1 + pow(frequency, 2);
    active(this) & this.tasks = 0 : 1;
    [activate] true : 2;
  }
}
```



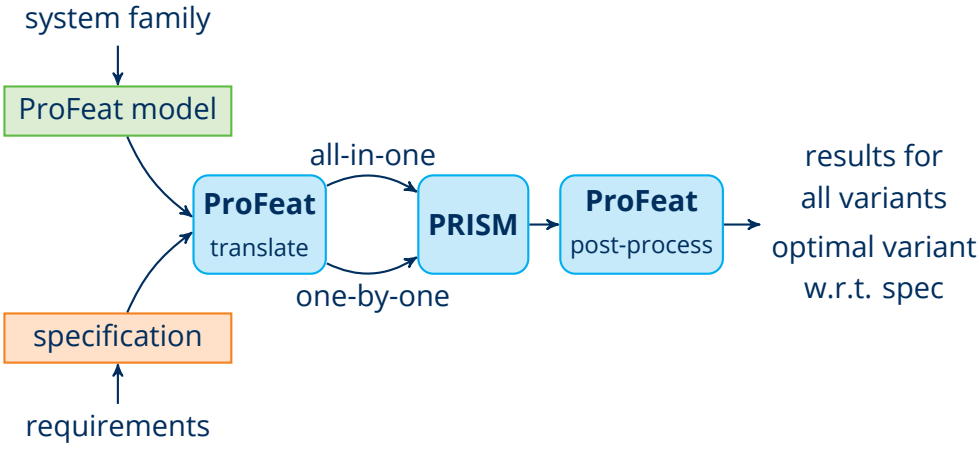
variant with `frequency=2`

Implementation: translational approach



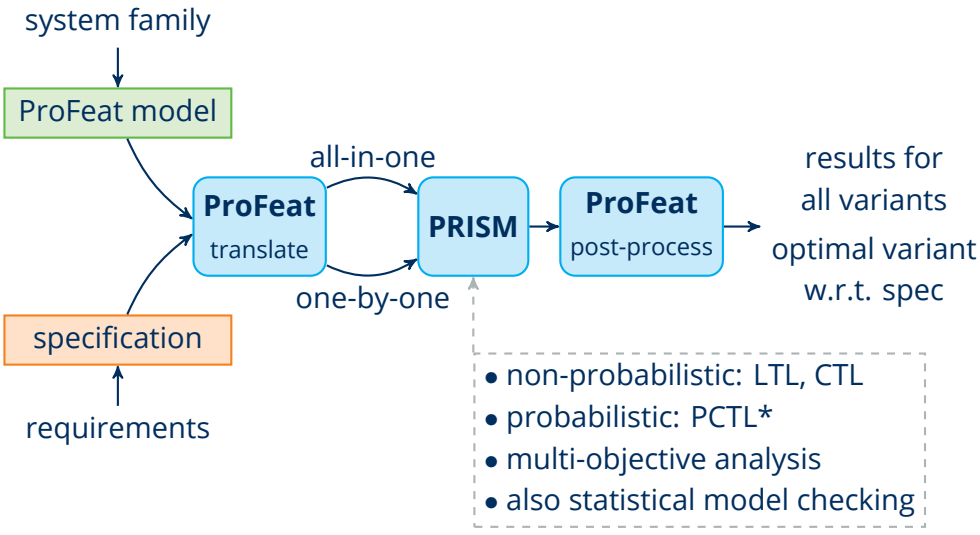
Philipp Chrszon, Clemens Dubslaff, Sascha Klüppelholz, Christel Baier.
ProFeat: Feature-oriented Engineering for Family-based Probabilistic Model Checking.
Formal Aspects of Computing 30(1), pp 45–75, 2018.

Implementation: translational approach

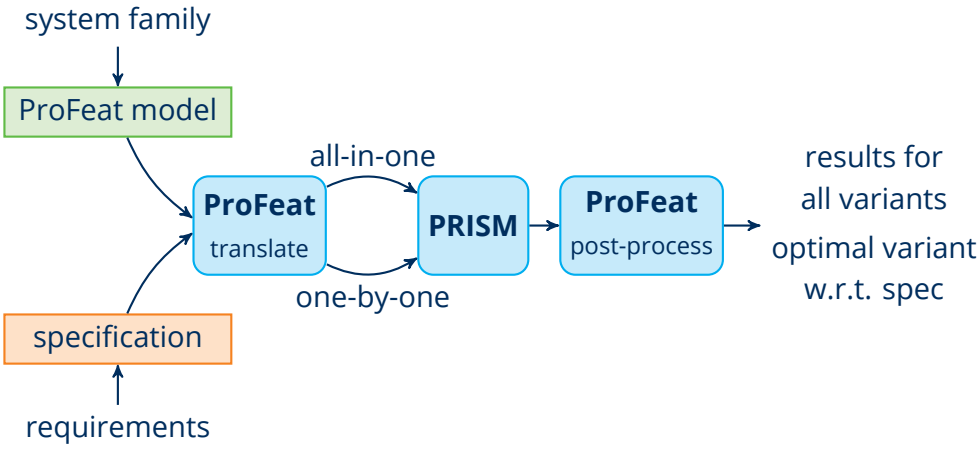


Philipp Chrszon, Clemens Dubslaff, Sascha Klüppelholz, Christel Baier.
ProFeat: Feature-oriented Engineering for Family-based Probabilistic Model Checking.
Formal Aspects of Computing 30(1), pp 45–75, 2018.

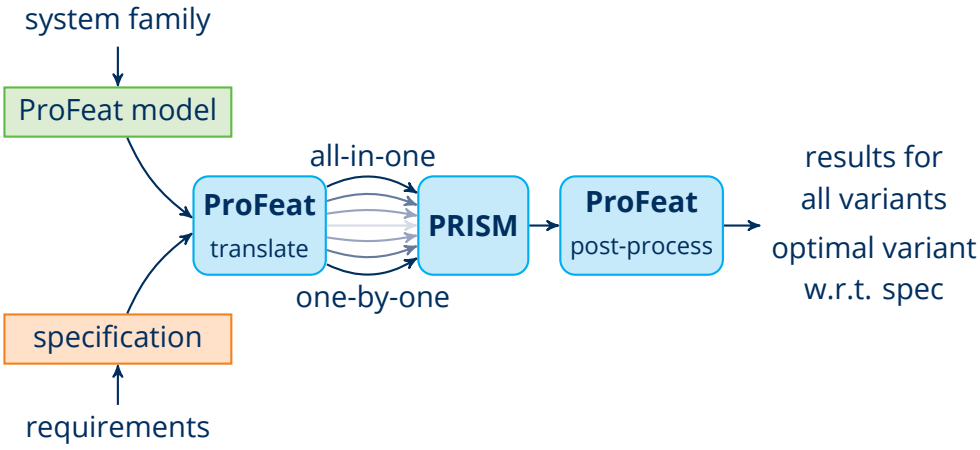
Implementation: translational approach



Implementation: translational approach



Implementation: translational approach



Analysis results: homogeneous vs. heterogeneous

Goal

Find the minimal expected energy and time for the variants.

Analysis results: homogeneous vs. heterogeneous

Goal

Find the minimal expected energy and time for the variants.

	tiles		minimal expected	
	#RISC	#DSP	energy	time
homogeneous	5	0	2.5 mJ	99.9 ms
heterogeneous	4	1	0.6 mJ	90.4 ms

Analysis results: strategy evaluation and synthesis

Goal

Find a scheduling strategy minimizing the expected energy or expected time for a certain job quota.

Analysis results: strategy evaluation and synthesis

Goal

Find a scheduling strategy minimizing the expected energy or expected time for a certain job quota.

strategy		minimal expected	
power	frequency	energy	time
nondet	nondet	62.0 mJ	90.4 ms
nondet	powersave	62.0 mJ	110.0 ms
nondet	performance	281.0 mJ	90.4 ms
always on	nondet	106.3 mJ	90.4 ms
always on	powersave	106.3 mJ	110.0 ms
always on	performance	599.8 mJ	90.4 ms

Conclusions

Summary

- ProFeat language for modeling families of stochastic systems using feature-oriented concepts
- Tool support for all-in-one, one-by-one or hybrid analysis
- Evaluation of system with heterogeneous tiled architecture

Conclusions

Summary

- ProFeat language for modeling families of stochastic systems using feature-oriented concepts
- Tool support for all-in-one, one-by-one or hybrid analysis
- Evaluation of system with heterogeneous tiled architecture

Future Work

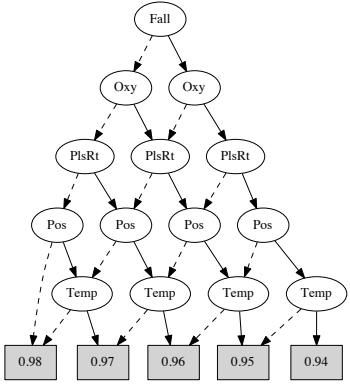
- Integration with other tools
- Extraction of feature-level knowledge from analysis results
- Visualization of analysis results?

Symbolic representation of analysis results

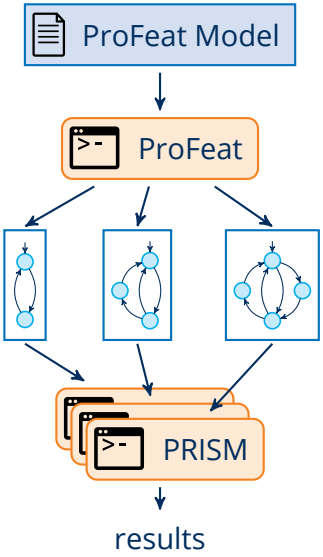
```
Final result: [0.9445746949695318,0.9792165174854354]
Results for initial configurations:
(Mem, Fall, Oxy, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.9445746949695318
(Mem, Oxy, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.953118529409139
(Mem, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.9617396442452253
(Mem, Pos, Temp, SACC, STemp)=0.9704387384917665
(Mem, PlsRt, SECG, STemp)=0.9792165174854354
               ... 295 lines omitted ...
```

Symbolic representation of analysis results

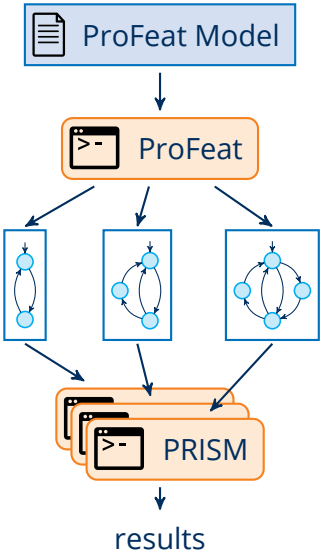
```
Final result: [0.9445746949695318,0.9792165174854354]
Results for initial configurations:
(Mem, Fall, Oxy, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.9445746949695318
(Mem, Oxy, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.953118529409139
(Mem, PlsRt, Pos, Temp, SACC, SSPO2, STemp)=0.9617396442452253
(Mem, Pos, Temp, SACC, STemp)=0.9704387384917665
(Mem, PlsRt, SECG, STemp)=0.9792165174854354
               ... 295 lines omitted ...
```

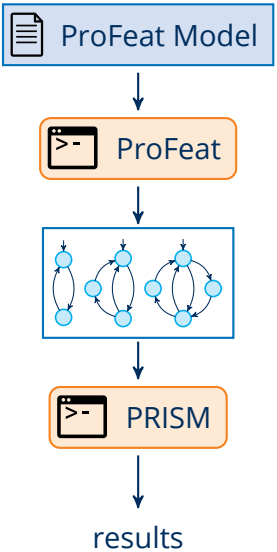


All-in-one and one-by-one analysis



All-in-one and one-by-one analysis





All-in-one and one-by-one analysis