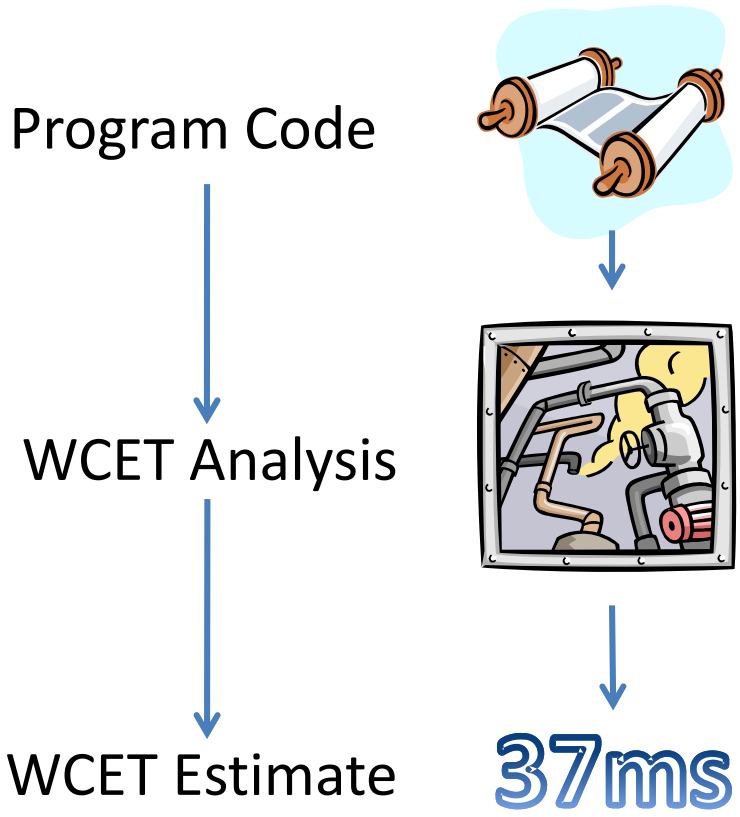
Interactive Timing Profiles based on Bayesian Networks

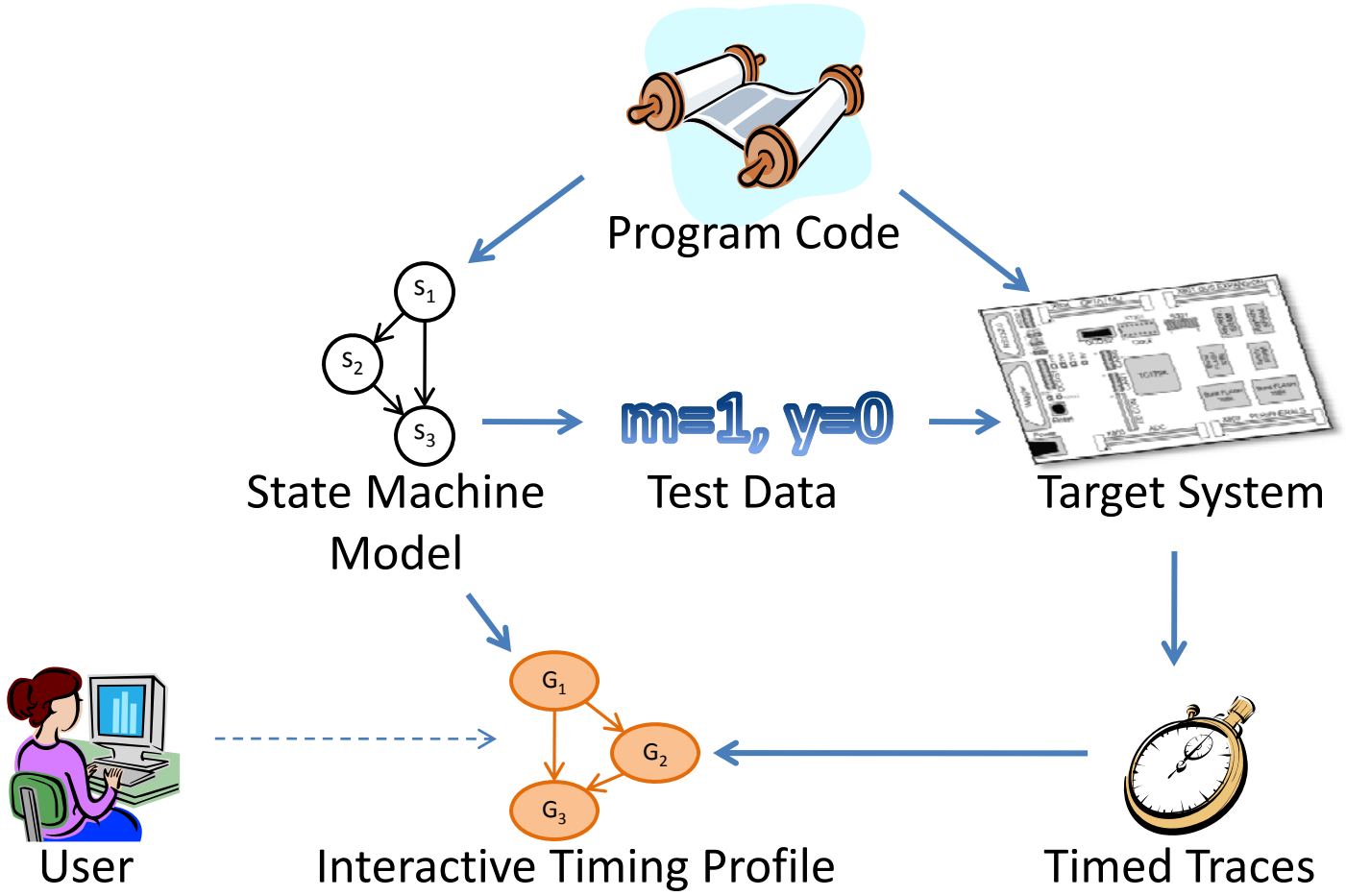
Michael Zolda, TU Wien

2008-07-01

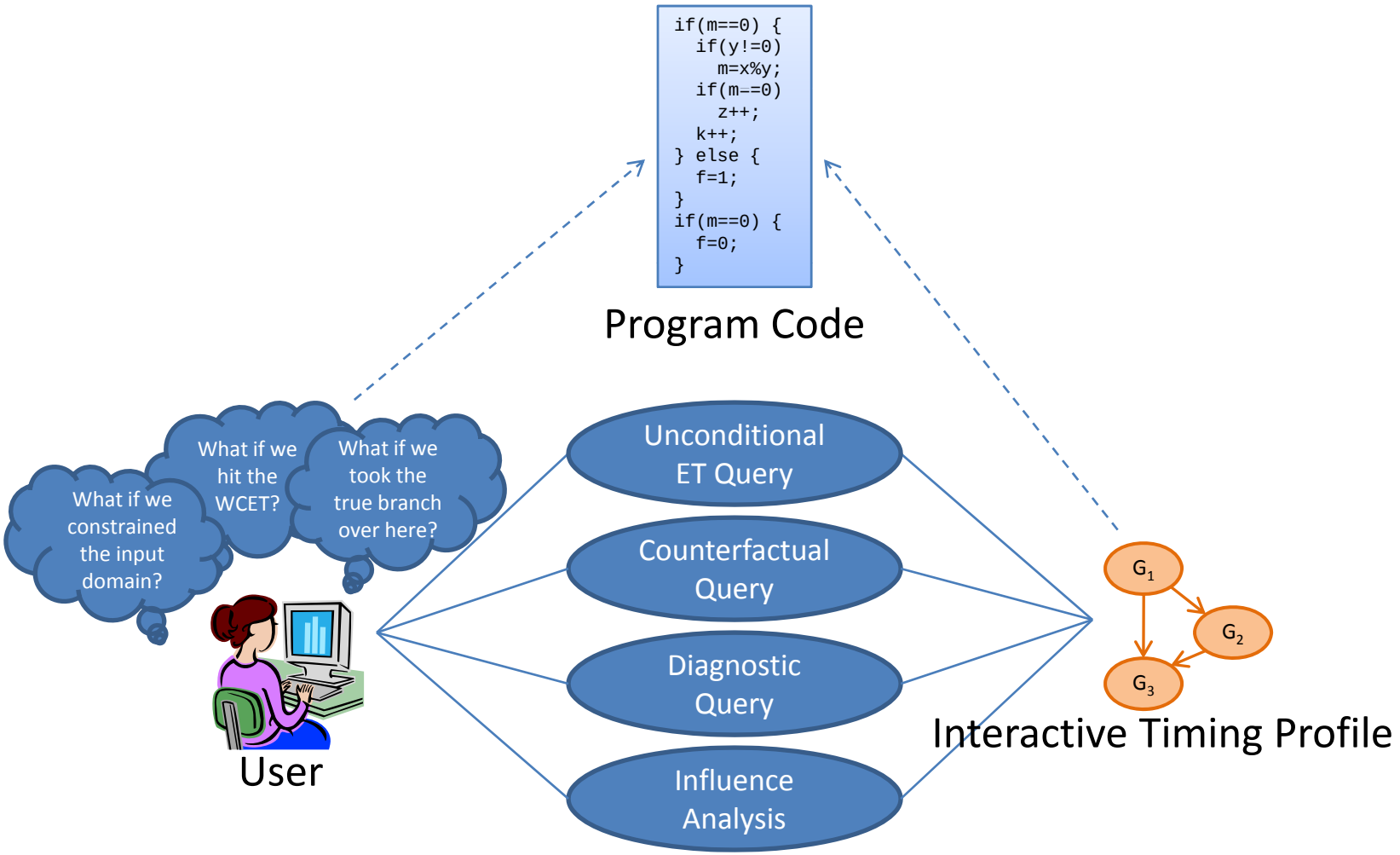
Traditional Workflow for WCET Analysis



Interactive Timing Profiles

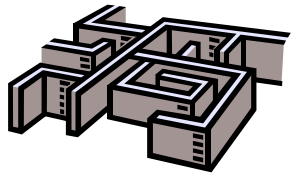


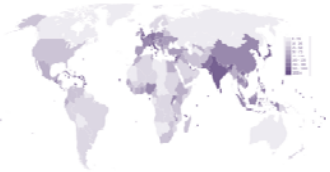
ITP User Interaction



Modeling with Bayesian Networks

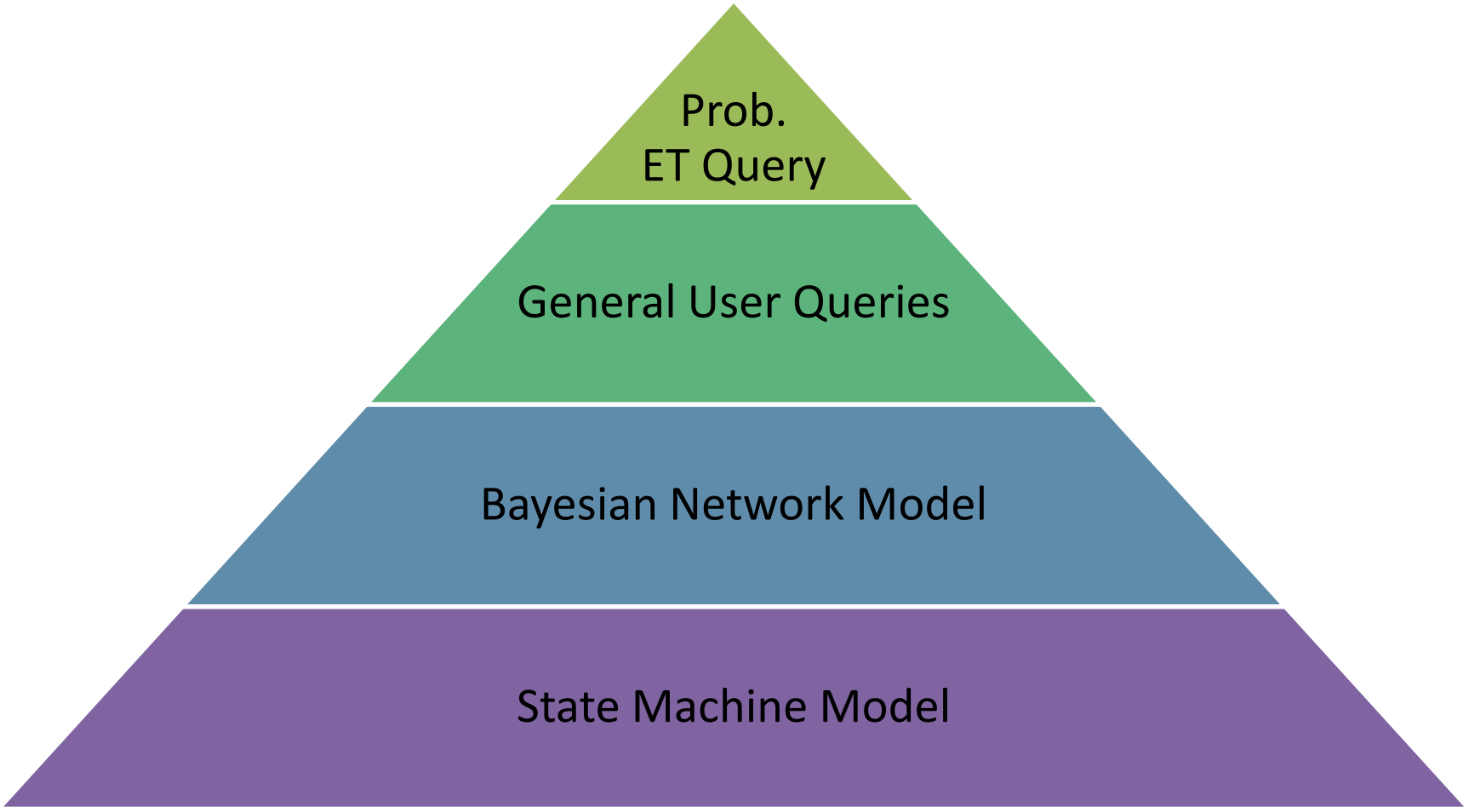
- Today‘s systems are becoming too complex
  - Software complexity, caches, pipeline, speculation, …
  - Can *neither* ignore nor model all the details
  - Use probabilistic model to summarize details



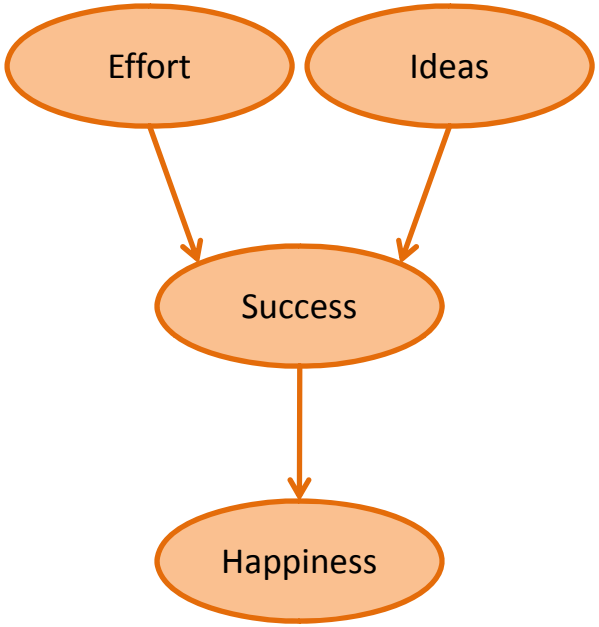


- Bayesian networks
  - Describe a probabilistic model
  - Random Variables ↔ Nodes
  - Conditional dependencies ↔ Arcs
  - Conditional Probability table for each node and its immediate predecessors

Approach Structure



Bayesian Network of Success



Effort		
	yes	0.5
	no	0.5

Ideas		
	yes	0.5
	no	0.5

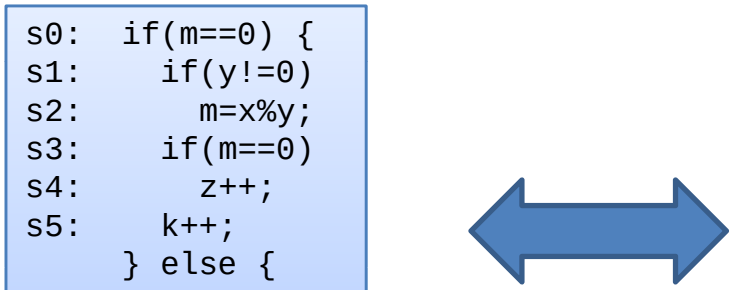
Effort		yes		No	
Ideas		yes	no	Yes	No
Success	yes	0.7	0.4	0.3	0.1
	no	0.3	0.6	0.7	0.9

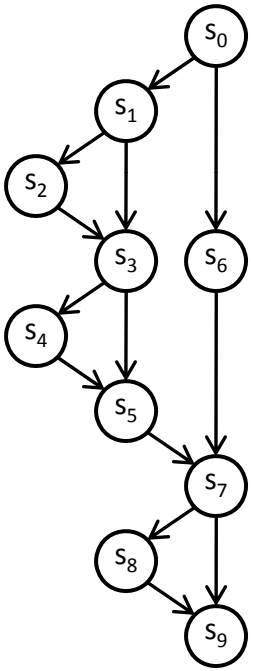
Success		yes	no
Happiness	yes	0.6	0.3
	no	0.4	0.7

- How likely is it that a happy person is putting in some effort?
  - Set evidence for variable *happiness*
  - Perform belief update
  - Read belief at variable *effort*

Abstract State Machine Model

```
s0:  if(m==0) {
s1:    if(y!=0)
s2:      m=x%y;
s3:    if(m==0)
s4:      z++;
s5:    k++;
     } else {
s6:    f=1;
     }
s7:  if(m==0) {
s8:    f=0;
s9:  }
```



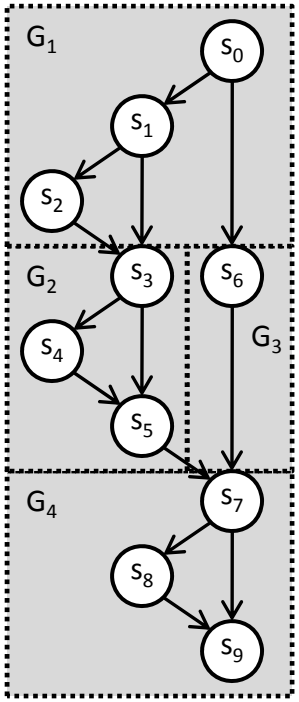


Program Code

State Machine Model

Segmentated Abstract State Machine Model

```
s0:  if(m==0) {
s1:    if(y!=0)
s2:      m=x%y;
s3:    if(m==0)
s4:      z++;
s5:    k++;
     } else {
s6:    f=1;
     }
s7:  if(m==0) {
s8:    f=0;
s9:  }
```



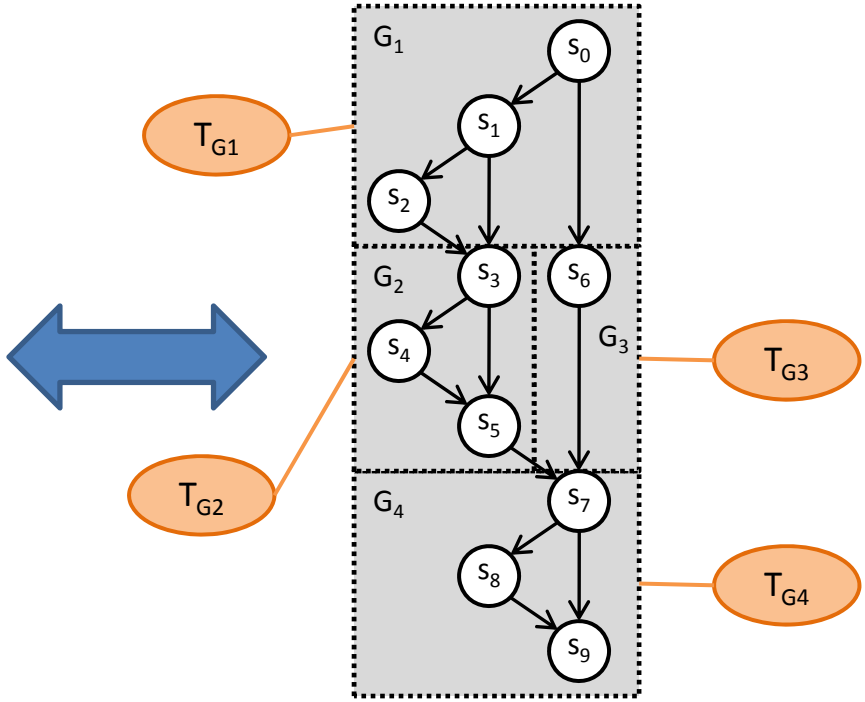
Program Code

State Machine Model

Segmentated Abstract State Machine Model

```
s0:  if(m==0) {
s1:    if(y!=0)
s2:      m=x%y;
s3:    if(m==0)
s4:      z++;
s5:    k++;
     } else {
s6:    f=1;
     }
s7:  if(m==0) {
s8:    f=0;
s9:  }
```

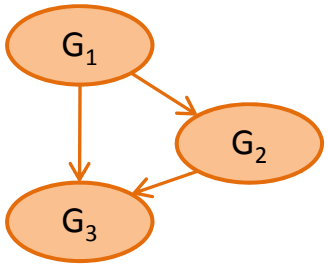
Program Code



State Machine Model

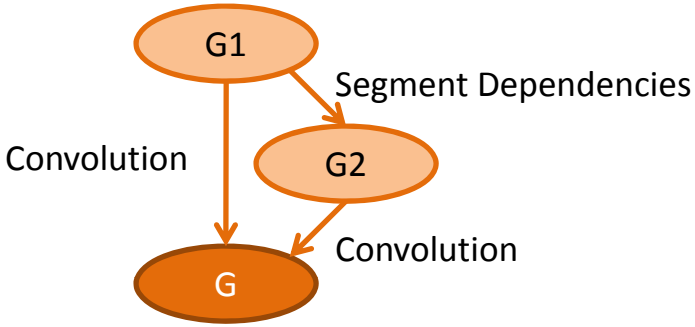
Deriving the ITP

- Deriving the structure
  - Context sets
  - Candidate segments
    - Knowledge about cache layout
    - Pipelining effects over segment boundaries
    - Control flow dependencies
- Deriving the parameterization
  - Classifying execution times
  - Use conditional relative frequencies



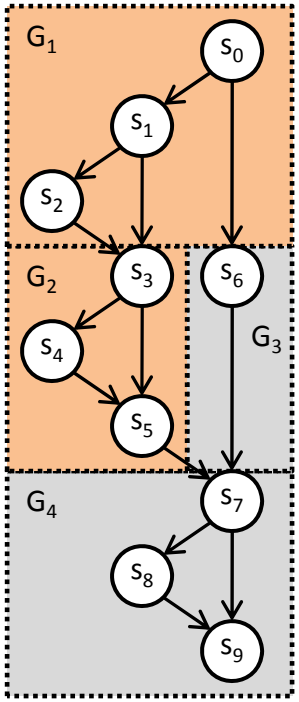
T_{G1}		10ms		11ms	
T_{G2}		20ms	21ms	20ms	21ms
T_{G3}	50ms	0.7	0.4	0.3	0.2
	51ms	0.3	0.6	0.7	0.9

Combining Execution Times



T_{G1}		10ms			11ms			null		
T_{G2}		20ms	21ms	null	20ms	21ms	null	20ms	21ms	null
T_G	30ms	1	0	0	0	0	0	0	0	0
	31ms	0	1	0	1	0	0	0	0	0
	32ms	0	0	0	0	1	0	0	0	0
	null	0	0	0	0	0	0	0	0	1
	inc.	0	0	1	0	0	1	1	1	0

CPT for Sequential Combination



State Machine Model

Future Work

- Segment concept
  - Useful segmentation
  - Identification of suitable segments
- Richer network structure
  - Timing information
  - Explicit modeling of control flow
  - Conditions on program variables
- Integration with the FORTAS model
- Outcome classification
- Implementation
- Quantitative results

Conclusion

- Approach for Timing Analysis
  - Interactive (What-if scenarios)
  - Probabilistic (Distributions of execution Times)
- WCET query is a special case

The End