

UPPAAL STRATEGO

Alexandre David Peter Gjør Jensen Kim Guldstrand Larsen
Marius Mikucionis Jakob Haahr Taankvist

Department of Computer Science, Aalborg University,
Selma Lagerlöfs Vej 300, 9220 Aalborg Øst, Denmark

October 30, 2014

Motivation - Different UPPAAL versions

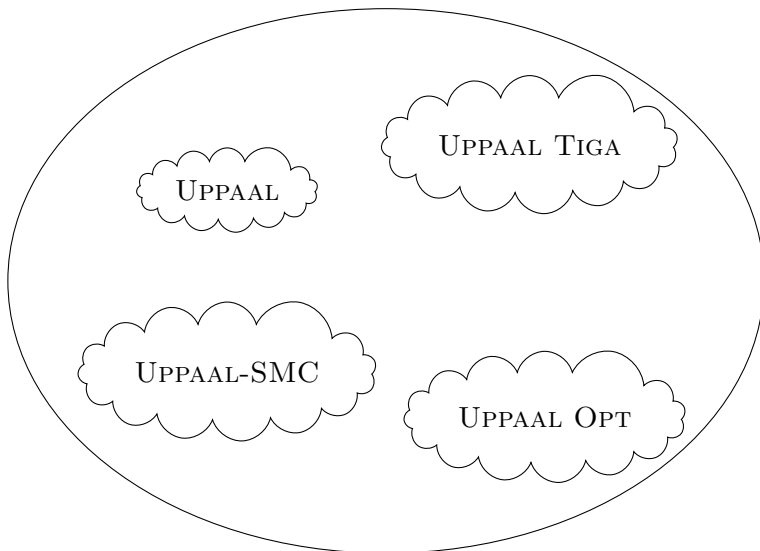
UPPAAL

UPPAAL TIGA

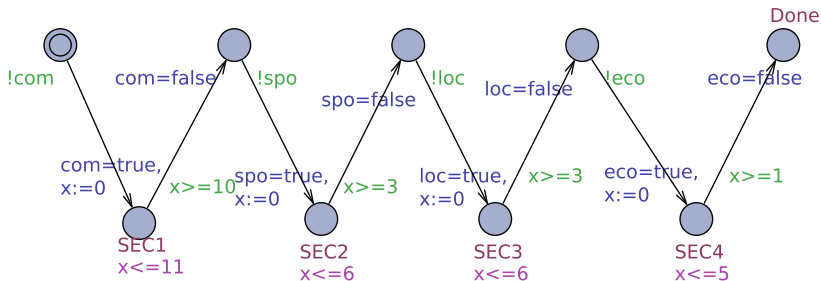
UPPAAL-SMC

UPPAAL OPT

Motivation - UPPAAL STRATEGO

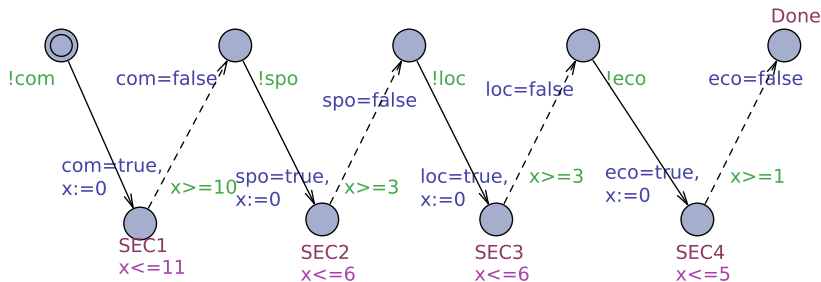


Branches of UPPAAL - UPPAAL



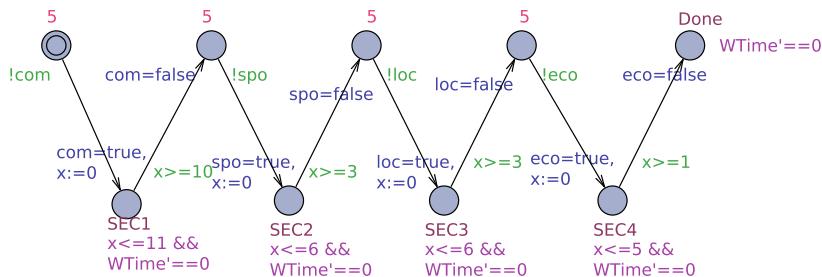
- Classical UPPAAL
- Classical timed model checking
- Timed Automata (TAs)
- Safety - $A[] \text{ not } (\text{Peter}.\text{SEC1} \ \&\& \ \text{Kim}.\text{Done})$
- Liveness -
 $A\langle \rangle \text{ Peter}.\text{Done} \ \&\& \ \text{Jakob}.\text{Done} \ \&\& \ \text{Kim}.\text{Done} \ \&\& \ \text{Marius}.\text{Done}$
- True/False answers

Branches of UPPAAL - UPPAAL TIGA



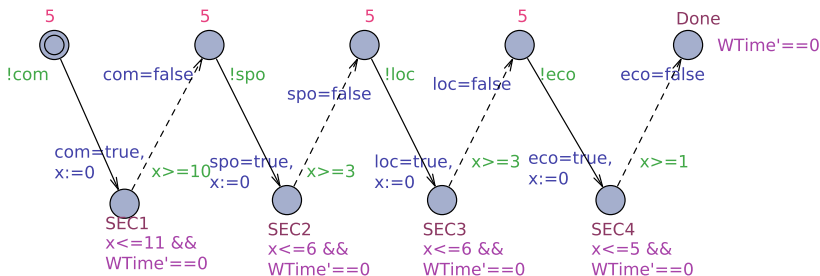
- Timed Games (TGAs) (Controller/Environment)
- Find a strategy
- Safety - $A[] \text{ Peter.Done} \ \&\& \ \text{Jakob.Done} \ \&\& \ \text{Kim.Done} \ \&\& \ \text{Marius.Done} \ \text{imply} \ \text{time} \leq 60$
- Strategies (most permissive)

Branches of UPPAAL - UPPAAL-SMC



- Stochastic timed automata.
- Probabilities -
 $\Pr[\leq 100] (\langle \rangle \text{Peter.Done} \ \&\& \ \text{Jakob.Done} \ \&\& \ \text{Kim.Done} \ \&\& \ \text{Marius.Done} \ \&\& \ \text{time} \leq 60)$
- Expectations
 $E[\leq 100, 1000] (\text{max: Peter.WTime})$
- Quantitative answers or True/False answers

Branches of UPPAAL - UPPAAL OPT

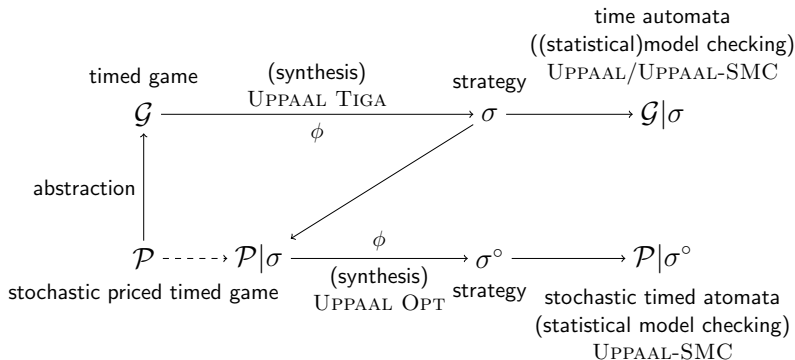


- $1\frac{1}{2}$ player games (Controller/stochastic environment)
- Learn a strategy
- Maximize/minimize
($\min E \text{ (Peter.WTime) } [\leq 100] : \langle \rangle \text{ Peter.Done}$)
- Near optimal strategies

Contribution

- Merge UPPAAL, UPPAAL TIGA, UPPAAL-SMC and UPPAAL OPT
- Make UPPAAL OPT accessible
- Facilitate exploration of “strategy-space”

Overview



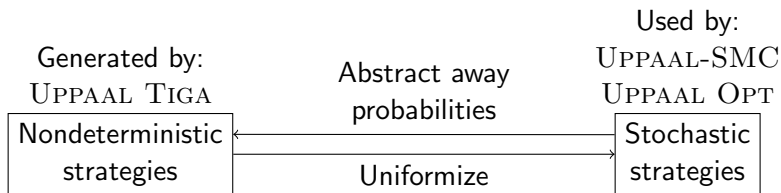
Types of Strategies

Generated by:

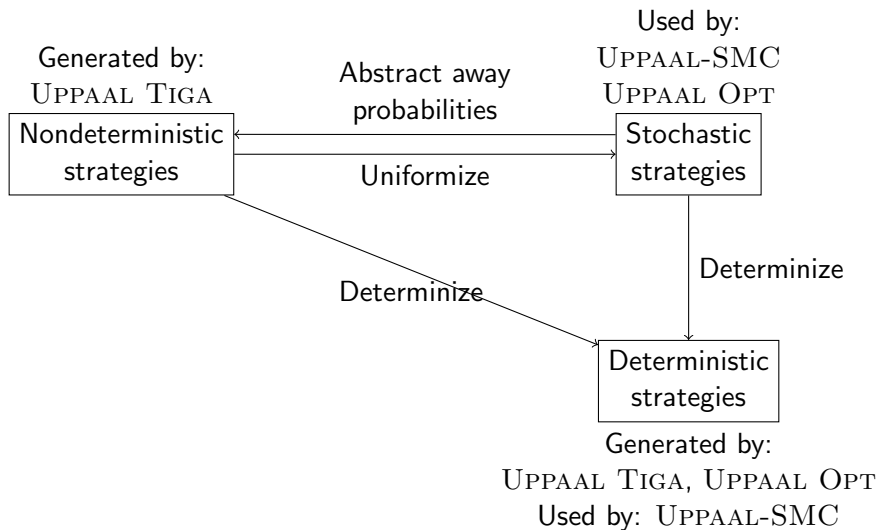
UPPAAL TIGA

Nondeterministic
strategies

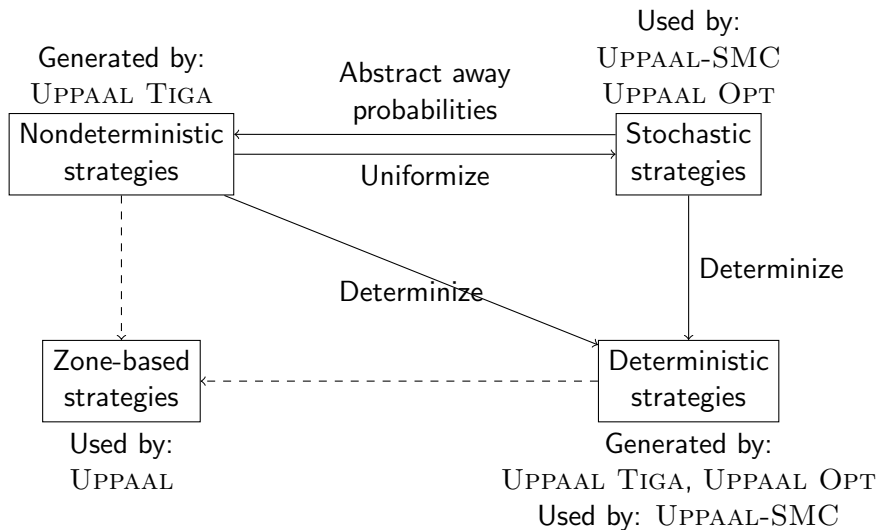
Types of Strategies



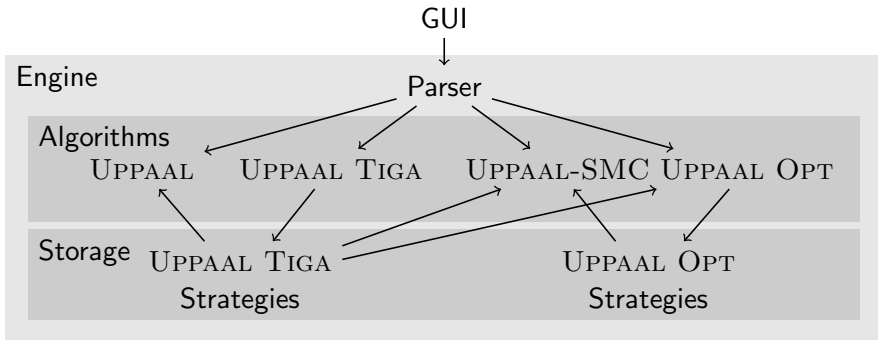
Types of Strategies



Types of Strategies



Architecture of UPPAAL STRATEGO



Updated Query Language

Strategy generators using UPPAAL OPT

Minimize objective	<code>strategy DS = minE (expr) [bound]: <> prop under NS</code>
Maximize objective	<code>strategy DS = maxE (expr) [bound]: <> prop under NS</code>

Strategy generators using UPPAAL TIGA

Guarantee objective	<code>strategy NS = control: A<> prop</code>
Guarantee objective	<code>strategy NS = control: A[] prop</code>

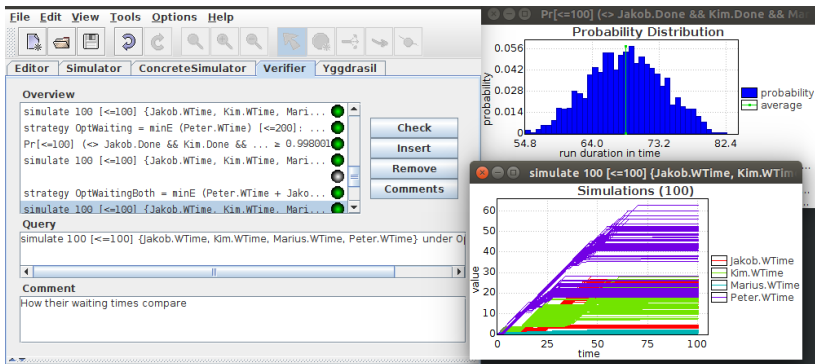
Statistical Model Checking Queries

Hypothesis testing	<code>Pr [bound] (<> prop) >= 0.1 under SS</code>
Evaluation	<code>Pr [bound] (<> prop) under SS</code>
Comparison	<code>Pr [bound] (<> prop1) under SS1 >= Pr [<=20] (<> prop2) under SS2</code>
Expected	<code>value E[bound;int](min: prop) under SS</code>
Simulations	<code>simulate int [bound]{expr1,expr2} under SS</code>

Model checking queries

Safety	<code>A[] prop under NS</code>
Liveness	<code>A<> prop under NS</code>

Demonstration



Conclusions and Further Work

- Conclusions
 - Integration of UPPAAL, UPPAAL-SMC, UPPAAL TIGA and UPPAAL OPT
 - New ways of exploring strategies
 - First time UPPAAL OPT is accessible
- Further work
 - Include other versions of UPPAAL
 - Play against learned strategies
 - Export/import strategies
 - More use of strategies (Model check of learned strategies)