

<b>Theorie reaktiver Systeme</b> <i>Theory of Reactive Systems</i>							Modulnummer:		
Bachelor Pflicht <input type="checkbox"/> Winf-Schwerpunkt-Pflicht <input type="checkbox"/> Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/> Winf-Wahl <input type="checkbox"/>				Schwerpunkt Computational Finance <input type="checkbox"/> E-Business <input type="checkbox"/> IT-Management <input type="checkbox"/> Logistik <input type="checkbox"/>					
Anzahl der SWS	V 2	UE 2	K 0	S 0	Prak. 0	Proj. 0	$\Sigma$ 4	Kreditpunkte: 6	Turnus i. d. R. angeboten alle 2 Semester
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Theoretische Informatik 1									
Vorgesehenes Semester: ab 1. Semester									
Sprache: Deutsch/Englisch									
Ziele: <ul style="list-style-type: none"> <li>• Semantische Alternativen für eingebettete Echtzeitsysteme bewerten können</li> <li>• Verständnis für die Grundkonzepte des Model Checkings entwickeln</li> <li>• Große (unendliche) Zustandsräume durch Abstraktion beherrschbar machen können</li> <li>• Semantische Modellierung zur Automatisierung bei Verifikation und Test einsetzen können</li> </ul>									
Inhalte: <ol style="list-style-type: none"> <li>1. Modelle der operationellen Semantik: Zustands-Transitionssysteme, markierte Transitionssysteme („Labelled Transition Systems LTS“), Markierte Transitionssysteme mit Zeit („Timed LTS“), Transitionssysteme mit Codierung der Refusal-Information – Finite State Machines (FSM) – Interleaving-Semantics versus „true Parallelism“ : Harel’s StepSemantik für Statecharts – Kripke-Strukturen</li> <li>2. Äquivalenz und Verfeinerung: Bisimilarität – Simulationsbeziehung - Verfeinerungen</li> <li>3. Fundamentale Modelleigenschaften: Deadlockfreiheit – Livelockfreiheit - Safety- und Liveness-Eigenschaften – Fairness</li> <li>4. Modell-orientierte Spezifikationsformalismen und ihre Semantik: Timed Automata – Hybrid Automata – Timed CSP</li> <li>5. Implizite Spezifikationsformalismen und ihre Semantik: Trace Logik mit und ohne Zeit – Temporallogiken: Linear Time Logic (LTL), Computation Tree Logic (CTL), Timed Computation Tree Logic (TTCL)</li> <li>6. Nachweis universeller Eigenschaften durch strukturelle Induktion über Syntax und operationelle Semantik.</li> <li>7. Modellprüfung</li> <li>8. Modellabstraktion</li> </ol>									
Unterlagen (Skripte, Literatur, Programme usw.): <ul style="list-style-type: none"> <li>• Edmund M. Clarke, Orna Grumberg and Doron A. Peled: “Model Checking”, The MIT Press, 1999</li> <li>• Christel Baier and Joost-Pieter Katoen: “Principles of Model Checking”, The MIT Press, 2008</li> <li>• K. Apt, E.-R. Olderog: “Verification of Sequential and Concurrent Programs”, Springer, 1991</li> </ul>									
Form der Prüfung: i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch oder mündliche Prüfung									
Arbeitsaufwand		Präsenz		56 h		Übungsbetrieb/Prüfungsvorbereitung		124 h	
		Summe		180 h					
Lehrende: Prof. Dr. J. Peleska					Verantwortlich: Prof. Dr. J. Peleska				