

Klassifikationsbäume in Echtzeit

Martin Leucker, Moritz Löser, Dirk Nowotka, Joachim Rischen



automatisierter Äquivalenzklassentest

Agenda

- Grundlagen
 - Klassifikationsbaummethode (CTM)
 - Zeit und Zustände
 - Matlab Testsystem
 - Äquivalenzklassentest Bosch Engineering
- Konzeption
 - Testprozess
 - Zeit und Zustände, Erweiterung CTM
 - Modellbildung
- Realisierung
 - Ausgangspunkt CTE
 - Programmiersprache
 - Implementierung
 - Anbindung an Testsystem
 - Demonstration
- Zusammenfassung
- Ausblick
 - Programm
 - Prozess



Klassifikationsbaummethode (CTM)

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Entwickelt von M. Grochtmann, K. Grimm (1993, Daimler-Benz AG)



- Visualisierung von Äquivalenzklassen (ÄK) und Testfällen
 - Bessere Übersicht
 - Leichtere Überprüfung von Tests
 - Testkomplexität leicht einschätzbar
- Hierarchische Struktur
 - Baumtiefe = feinere Zerlegung
 - Verschiedene Testphasen einheitlich darstellbar
 - Grobe Kundenanforderung → einzelne Signalwerte
 - Reine Funktionalität → Zeitverhalten



Universität Stuttgart

3



BOSCH
Bosch Engineering GmbH

Klassifikationsbaummethode (CTM)

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

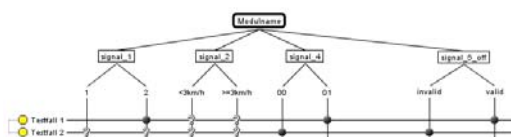
- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Integrierte Darstellung der Ausgangsgrößen möglich
- Testfall: Markierung der Klassen



- Classification Tree Editor (CTE) weit verbreitetes Tool zur
 - Baumerstellung
 - Testfallerzeugung
 - Testdokumentation



Universität Stuttgart

4



BOSCH
Bosch Engineering GmbH

Systemzustand

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

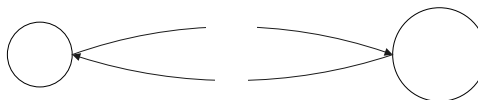
- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Das zu testende System besitzt einen „Zustand“
 - „Zustand“ impliziert frühere Eingabedaten
- Testfall:
 - Abbildung: Eingabewerte $\times$ Randbedingungen $\times$ Zustand $\rightarrow$ Sollzustand
- Modell für Systeme mit Zustand: Automat
 - System mit endlicher Anzahl an Zuständen
 - $\rightarrow$ endlicher Automat



Universität Stuttgart

5



BOSCH
Bosch Engineering GmbH

Zeitautomat

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

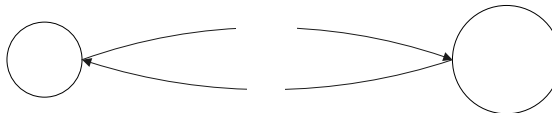
- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Für Echtzeitsysteme ist endlicher Automat nicht hinreichend
- Erweiterung des endlichen Automaten durch Uhren
 - beliebig viele Uhren möglich
 - Uhren können separat zurückgesetzt ($U_1 := 0$) werden
 - Zustände und Zustandsübergänge müssen Uhrenbedingungen genügen



- Worte $\rightarrow$ Zeitworte
 - Paare von Wert und Dauer
 - z.B. geschlossen, 1h $\rightarrow$ offen, 9s $\rightarrow$ geschlossen, 3 Tage ...



Universität Stuttgart

6



BOSCH
Bosch Engineering GmbH

Bosch Engineering /Matlab Testsystem

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

→ XML/Matlab Testschnittstelle zu verschiedenen Prüflingen

- Simulink Modelle
- Ausführbarer C-Code (OOL_EXE)
- Ascet Modelle

→ Wichtige Bestandteile eines Testfalls

- Stimulation: Signaldatei für sämtliche Eingaben
- Manipulation: Matlab-Code um Signalwerte zu ändern
- Analyse: Matlab-Code zum Vergleich von Ist- und Sollwerten an den Ausgängen des Prüflings

→ Ausgabe Testergebnis

- Ausführlicher HTML Report
- Signaldatei der Ausgabe des Prüflings

→ Bietet Schnittstelle zu MS-Excel als Frontend



Universität Stuttgart

7



BOSCH

Bosch Engineering GmbH

Bosch Engineering / Äquivalenzklassentest

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

→ Im Modultest (V-Modell: „Komponententest“) eingesetzt

→ Ein- und Ausgabegrößen sind in UML Architektur spezifiziert

→ ÄK werden anhand von Anforderungen und Architektur gebildet

→ Da „Modul“ sehr nah am Code: Relation zu Whitebox Test

- Ein- und Ausgabegrößen $\approx$ Variablen
- Kombination von Eingabewerten $\approx$ Anforderung $\approx$ Programmpfad
- Test aller Anforderungen + fehlende Anforderungen + Pfade im toten Code $\approx$ 100% Pfadüberdeckung
- Testfälle für jede Kombination von ÄK (vollständiger ÄK-Test) + Pfade im toten Code $\approx$ 100% Pfadüberdeckung



Universität Stuttgart

8



BOSCH

Bosch Engineering GmbH

Bosch Engineering / Äquivalenzklassentest

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Vollständiger Test: Testfall für jede Kombination von ÄK
- Kompromiss: Test jeder Anforderung für jede Kombination der restlichen Signale
- Nachteile
 - Testdauer entsprechend der Anzahl der Kombinationen
 - Keine Trennung zwischen
 - Pfadüberdeckung aus Anforderungen
 - Pfadüberdeckung aus Kombination der ÄK
- Vorteile
 - Hohe/höhere Pfadüberdeckung
 - Gleichbleibende Anzahl an Testfällen
 - Falls eine Anforderung nicht für jede Kombination erfüllt ist
 - Fehlende Anforderung, d.h. gewünschter aber nicht dokumentierter Einfluss einer Eingabegröße auf Sollverhalten
 - Fehler, d.h. nicht gewünschter Einfluss einer Eingabegröße auf Sollverhalten
 - Pfadüberdeckung < 100% weist auf toten Code hin



automatisierter Äquivalenzklassentest, Anforderungen

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Möglichst vollständige Testspezifizierung im CTE
 - Klassenbaum: Testdokumentation, Stimulierzeugung
 - Testfallspezifizierung und Testfallerzeugung für Matlabtestsystem
 - Einbeziehung von Zeit und Zuständen
- Für vollständige Integration des CTE ist neuer Testprozess nötig
- Für Abbildung von Zeit und Zuständen muss CTM erweitert werden



automatisierter Äquivalenzklassentest, Prozess

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

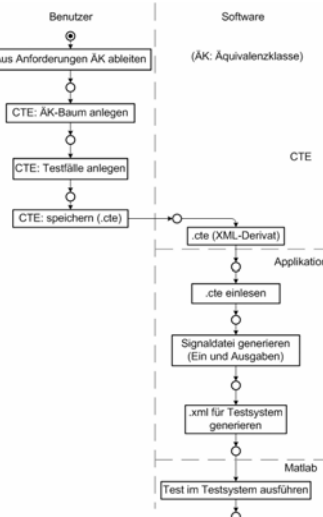
Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess



Universität Stuttgart

11



BOSCH
Bosch Engineering GmbH

Zeit und Zustände, Erweiterung CTM (Mirco Conrad 2004)

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

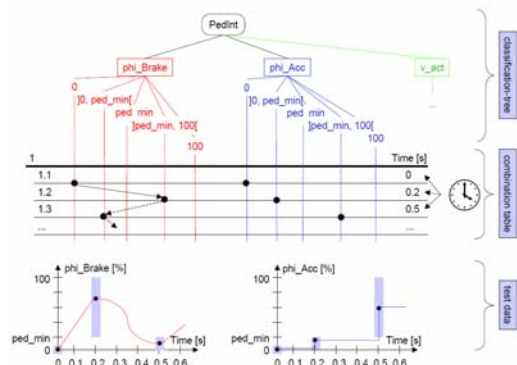
- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Zeit und Zustände bisher dargestellt über Testfallsequenzen (CTM/ES)
- nicht im Klassifikationsbaum
 - schlechtere Übersichtlichkeit
 - korrektes Timing der Signale untereinander muss gewährleistet



Universität Stuttgart

12



BOSCH
Bosch Engineering GmbH

Zeit und Zustände, alternative Erweiterung CTM

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

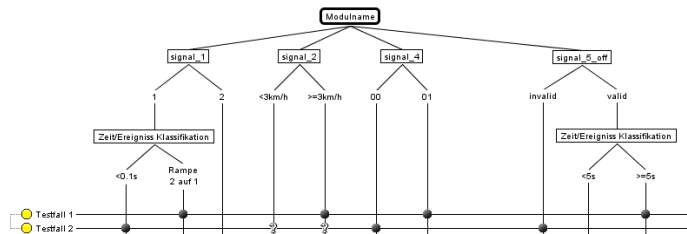
- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Vorschlag: alternative Darstellung von Zeit und Zuständen
 - integriert in Klassifikationsbaum



- Ggf. zusätzliche Ebene für Zeit und Zustände
 - Übersicht bleibt weitgehend erhalten
 - einheitliche Darstellung aller ÄK
- korrektes Timing übernimmt Modell im Hintergrund



Universität Stuttgart

13



BOSCH
Bosch Engineering GmbH

Modellbildung: Signal → Zeitautomat

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Warum Abbildung auf Zeitautomaten?
 - Sowohl Systemzustände als auch Echtzeitverhalten modellierbar
 - Gut erforschte Theorie, inkl. formaler Verifikation
 - Auf Modulebene ist Komplexität (Zustandsraum) überschaubar
- Abbildung:
 - Jedes Eingangssignal bildet einen Zeitautomaten
 - ÄK → Zustand
 - Startzustände: jeder Zustand ohne Zeitbedingung
- Ausgangssignale werden modelliert wie Eingangssignale
- Modellverifikation:
 - Pro Ausgangssignal wird ist-Zustand mit soll-Zustand verglichen



Universität Stuttgart

14



BOSCH
Bosch Engineering GmbH

Beispiel (aus „Backup Brake Function“)

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

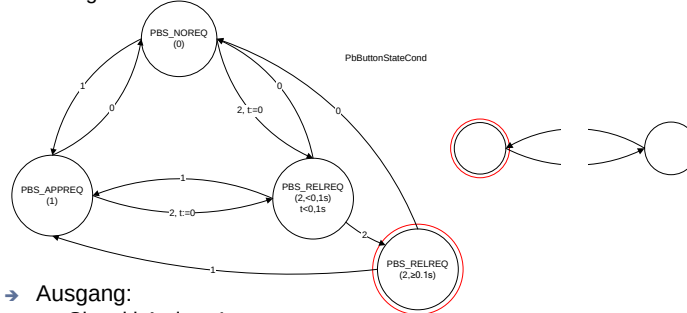
Zusammenfassung

Ausblick

- Programm
- Prozess

→ Eingang:

- Signal PbButtonStateCond=2 für mindesten 0,1s
- Signal vVeh<3km/h



→ Ausgang:

- Signal bActive=1



Randbedingungen

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

→ Beliebiger (rechteckiger) Signalverlauf darstellbar

- Für Ein- und Ausgangssignale
- dargestellt durch Zeitworte

→ Für jedes Signal mit Zeitbedingung muss ein Wert ohne Zeitbedingung existieren

- „Füllwert“ für Synchronisation

→ Komplexitätsgrenze

- Anzahl der Zustände für Modul mit 10 Eingängen und jeweils 10 ÄK: 10^{10}
- Testfälle mit zu vielen freien ÄK müssen ggf. in mehrere Testfälle aufgeteilt werden



Ausgangspunkt CTE

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Klassenbaum und Markierung der Klassen nicht hinreichend für Testdurchführung
 - Zusätzliche Attribute für die meisten Elemente notwendig z.B.
 - Repräsentant einer Klasse
 - Trennung zwischen Ein- und Ausgängen
- CTE besitzt Möglichkeit, Elementen „Tags“ zu geben für
 - Attribute
 - Anforderungen
- CTE speichert in XML Derivat



Java

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Relativ modern
 - Gute Möglichkeiten der Strukturierung (Kapselung, OO)
- Sehr gute XML Unterstützung
 - Sowohl Quell- (CTE) als auch Zielformat (Matlab Testsystem) sind XML Derivate
- CTE besitzt Java API
 - Leider nicht Dokumentiert
- Matlab Datei (.mat) Schnittstelle vorhanden
 - Einfache Anbindung an Matlab Testsystem



Implementierung, 2 Algorithmen

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Kombination aller valider Äquivalenzklassen
 - Algorithmus, der das kartesische Produkt berechnet
 - Ursache der enormen Speicherkomplexität ($\approx$ Kardinalität des kartesischen Produkts)
- Korrekte (synchrone) Kombination zeit- und zustandsbezogener Klassen
 - Jede ÄK-Kombination braucht gleiche Länge (Dauer)
 - Zum Auffüllen dient erwähnter Füllwert



Anbindung an Matlab Testsystem

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Ausgabe einer „T-spec.xml“ pro Testfallauswahl
- Ausgabe einer Signaldatei (Stimuli) pro Testfall
- Resultatssignal für jede Analyse (0=„failed“, 1=„passed“)
 - ..._result



Demonstration

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Modul BBF (Backup Brake Function) aus APB Projekt
- Durchschnittliche Komplexität
- Besitzt zeitbezogene Äquivalenzklassen



Zusammenfassung

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Integration von Matlab Testsystem und CTE
 - Spezifizierung und Durchführung von Äquivalenzklassentests
 - Berücksichtigung von Zeit und Zuständen
- Prozessanpassung
 - CTE ersetzt Excel als Benutzerschnittstelle
 - Weniger manuelle Eingriffe
 - Keine Matlabprogrammierung nötig /nicht mehr möglich
- Modell
 - Zeitautomaten
- Javaapplikation, Kommandozeileninterface



Ausblick: Programm

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Verbesserung der Benutzerschnittstelle
 - Grafische Benutzerschnittstelle
- Einbeziehung weiterer Daten
 - Selbständige Repräsentantenwahl (z.B. Grenzwertanalyse)
- CTE API nutzbar machen
- oder neuen CTE entwerfen



Ausblick: Prozess

Grundlagen

- Klassifikationsbaum-methode (CTM)
- Zeit und Zustände
- Äquivalenzklassentest
- Bosch Engineering

Konzeption

- Testprozess
- Erweiterung CTM
- Modellbildung

Realisierung

- Ausgangspunkt CTE
- Programmiersprache
- Implementierung
- Anbindung an Testsystem
- Demonstration

Zusammenfassung

Ausblick

- Programm
- Prozess

- Formale Formulierung der Anforderung
 - Schließt Lücke zu formaler Modelverifikation
 - Es existieren Sprachen die Echtzeitverhalten modellieren
 - Z.B. Timed Computation Tree Logic (TCTL)
 - automatische Erzeugung der ÄK?



→ Fragen?



→ Danke für Ihr Interesse!

