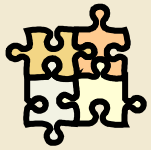
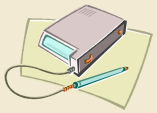
Verifikation regelbasierter Konfigurationssysteme

… making an automaton play any game (depends) upon the possibility of the machine being able to represent all the ***myriads of combinations*** relating to it.

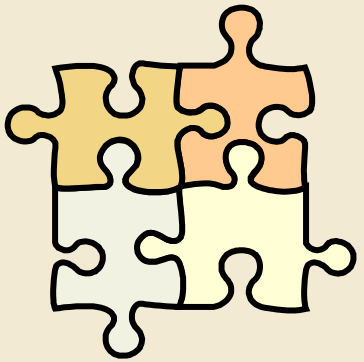
Charles Babbage, “Passages from the Life of a Philosopher ”, 1864

Carsten Sinz — Promotions-Kolloquium — 17.12.2003

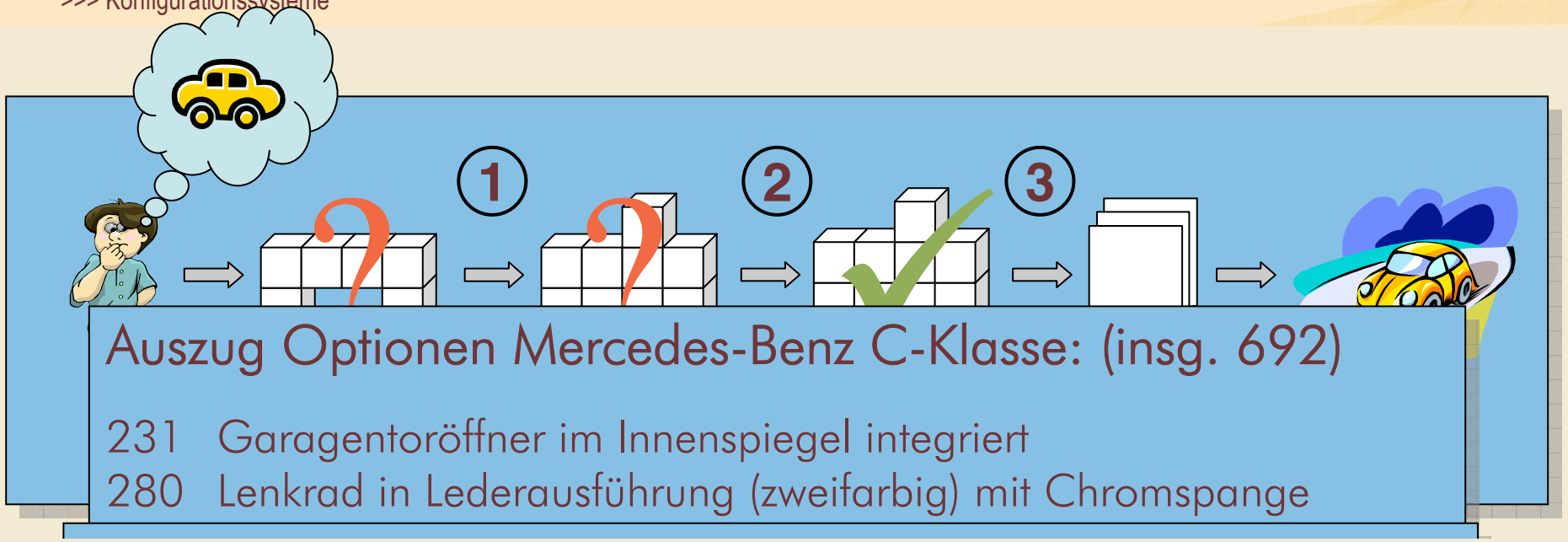
Übersicht

- Konfigurationssysteme 
- Verifikationsprozess 
- Empirische Feststellungen 

Teil 1: Konfigurationssysteme



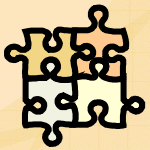
Konfiguration - Hintergrund



Formalismus erforderlich zur

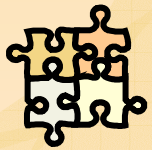
- Darstellung der Kombinationen und Restriktionen
- Steuerung des Bearbeitungsprozesses

Regelbasierter Formalismus



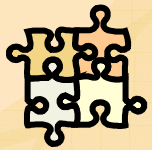
- Situations-Aktions-Regeln (if ... then ...)
  - liefern „natürliche“ Formulierung von Expertenwissen
  - beschreiben Produktstruktur, Restriktionen und Transformationen
- Zusammenhänge
  - Bestellcodes ≅ Boolesche Variablen
  - Situationen ≅ aussagenlogische Formeln
  - Aktionen ≅ Setzen von Booleschen Variablen auf true/false
- Auswertungsalgorithmus steuert Regelauswahl
  - Prüfung einer Konfiguration ≅ Berechnungslauf des Regelsystems
  - Semantik der Regeln und des Auswertungsalgorithmus aber häufig nicht klar spezifiziert

Aufgabenstellung und Zielsetzung



- Ausgangspunkt
  - Bestehendes regelbasiertes Konfigurationssystem DIALOG (Mercedes-Benz PKW und LKW)
  - keine formale Semantik der Regeln (Datenbank mit aussagen-logischen Formeln, Auswertung durch COBOL-Programm)
- Zielsetzung
  - Prüfung des Regelsystems, Verbesserung der Datenqualität
  - Unterstützung bei Wartungsarbeiten
- Notwendige Schritte
  1. Exakte Festlegung der Prozess- und Regelsemantik
  2. Erstellung von Korrektheitskriterien
  3. Automatische Prüfung dieser Kriterien

Konfig.-Regeln Mercedes-Benz



- Zusteuerungsregeln: **if Z_i then add code c_i**

  Fügen Bestellcodes c_i bedingt dem Auftrag hinzu

- Baubarkeitsregeln: **if $\neg(c_i \Rightarrow B_i)$ then "error"**

  Prüfen Bedingung für im Auftrag vorhandenen Code c_i

- Teileauswahlregeln: **if T_j then select part p_j**

  Fügen Teil p_j dem Auftrag bedingt hinzu

((-L/(M111+M23+M001/M112+M28/M113)+-(220/248/289/331/480/481/500/540/611/656/657+956/819/875+-(460/M113)/882/W10/Y94/Y95/X35/X59/X62))+-R)+((-L/M113+-X62/M112+M28+-(772/774/X62)/M111+M23+M001+-(280+-460/772/774/X62))+-R)+((-L/M112+M28+222+223+231+254+292+423+(460/249+461+551+810)+(524+668+634+636/820)+543+581+679+(955+265+657+(140A/200A)/956+570+(201A/208A))+809/M112+M28+221+222+231+254+292+(349/460)+423+(460/249+461+551+810)+(524+668+634+636/820)+543+581+679+955+265+657+(140A/200A)+800/M112+M28+221+222+231+254+292+(349/460)+423+(460/249+461+551+810)+(524+668+634+636/820)+543+581+679+956+570+(201A/208A)+800/M113+231+249+254+265+441+(460/461)+(551/460)+(810/460)+(524+668+634+636/820)+543+580A+809/M113+231+249+254+265+(349/460)+441+(460/461)+(551/460)+(810/460)+(524+668+634+636/820)+543+580A+800/M111+M23+M001+221+231+249+254+292+423+(524+634+636+668/820)+461+543+551+580+673+679+(955+265+657+(140A/200A)/956+570+(201A/208A))+809/M111+M23+M001+231+249+254+292+423+(524+634+636+668/820)+460+543+580+673+570+040+584+(140A/080A)+809/M111+M23+M001+231+249+254+292+423+(524+634+636+668/820)+460+543+580+673+570+955+221+657+679+(140A/200A)+809/M111+M23+M001+231+249+254+292+423+(524+634+636+668/820)+460+543+580+673+570+956+221+679+(201A/208A)+809/M111+M23+M001+221+231+249+254+292+(349/460)+423+(524+634+636+668/820)+461+543+551+580+673+679+955+265+657+(140A/200A)+800/M111+M23+M001+221+231+249+254+292+(349/460)+423+(524+634+636+668/820)+461+543+551+580+673+679+956+570+(201A/208A/258A)+800/M111+M23+M001+231+249+254+292+(349/460)+423+(524+634+636+668/820)+460+543+580+673+570+040+584+(140A/080A)+800/M111+M23+M001+231+249+(349/460)+254+292+423+(524+634+636+668/820)+460+543+580+673+570+955+221+657+679+(140A/200A)+800/M111+M23+M001+231+249+254+292+(349/460)+423+(524+634+636+668/820)+460+543+580+673+570+956+221+679+(201A/208A/258A)+800)+-R)+L+(-(W02/W03/W04/W05/W07/W08/W11/W43/W44/W45/Y93))+(-(202/205/206/207/208/209/210/228/230/236/243/270/273/278))+(-(304/312/315/316/317/318/328/330/333/342/343/347/350/351/354/356))+(-(450/482/490/498))+(-(533/550/561/562/565/592))+(-(613/617/623/625/630/640/643/645/648/653/655/665/675/676/681/682/687/693))+(-(702/703/704/706/707/715/718/724/750/752/753/754/755/756/759/776/783/791/793))+((823/825/826/827/828/829/830/831/832/833/834/835/836/837/838/839/852/853/854/855/884))+(-(913/918/925/931/932/935/937/938/947/948/959/960/970/974/975/979/980/981/982/984/986/987/988/994/997))+(-(X02/X03/X04/X05/X07/X08/X09/X11/X22/X23/X24/X25/X26/X31/X33/ X34/X51/X52/X54/X55/X57/X58/X60/X61/X63/X64))

-(420/42 /424/426/429/913)

until nc ult

for i=1 to n do

if ¬($c_i \Rightarrow B_i$) then “error”

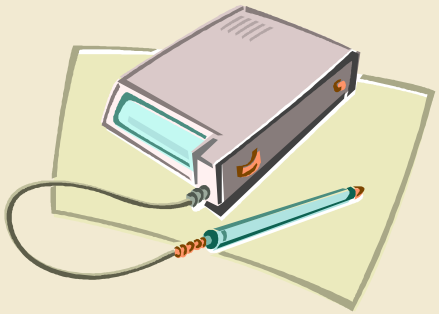
B *Baubarkeits-prüfung*

for j=1 to k do

if T_j then select part p_j

T *Teilebedarfs-ermittlung*

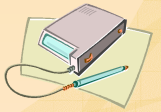
Teil 2: Verifikation



Warum Verifikation?

- Erstellung und Wartung des Regelsystems schwierig und fehleranfällig
- Fehler können gravierende Auswirkungen haben:
  - Inkorrekte Klassifikation von Aufträgen
  - Überflüssige Teile in der Stückliste
  - Fehlerhafte Teilebedarfsermittlung
- Hohe Änderungsrate des Regelsystems

Festlegung der Prozess-Semantik

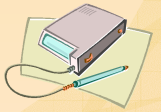


- Sprache: **Dynamische Aussagenlogik (PDL)**
  - Multimodale Logik, „logic of programs“
  - Erlaubt natürliche Darstellung imperativer Programme
  - Algorithmenspezifikation anhand von Nachbedingungen:

    [P]E „Nach allen Ausführungen von Programm P gilt Eigenschaft E.“

    ⟨P⟩E „Es gibt einen Lauf von Programm P, nach dem Eigenschaft E gilt.“
  - Programme P sind reguläre Ausdrücke über atomaren Programmen
- Regeln, Steuerungsalgorithmus ≅ PDL-Programme

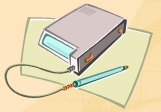
Korrektheitskriterien



1. Keine unzulässigen Bestellcodes
2. Konsistenz der Zusteuerung
3. Keine überflüssigen Teile
4. Keine Mehrdeutigkeiten in der Stückliste

… und einige weitere.

Unzulässige Bestellcodes



Def.: Bestellcodes, die in keinem gültigen Auftrag vorkommen können, heißen unzulässig.

- Konsequenz: Ein unzulässiger Bestellcode kann vom Kunden nicht wirklich gewählt werden
- Formalisierung in PDL:

Code c unzulässig gdw. $c \Rightarrow [Z;B]\mathsf{false}$

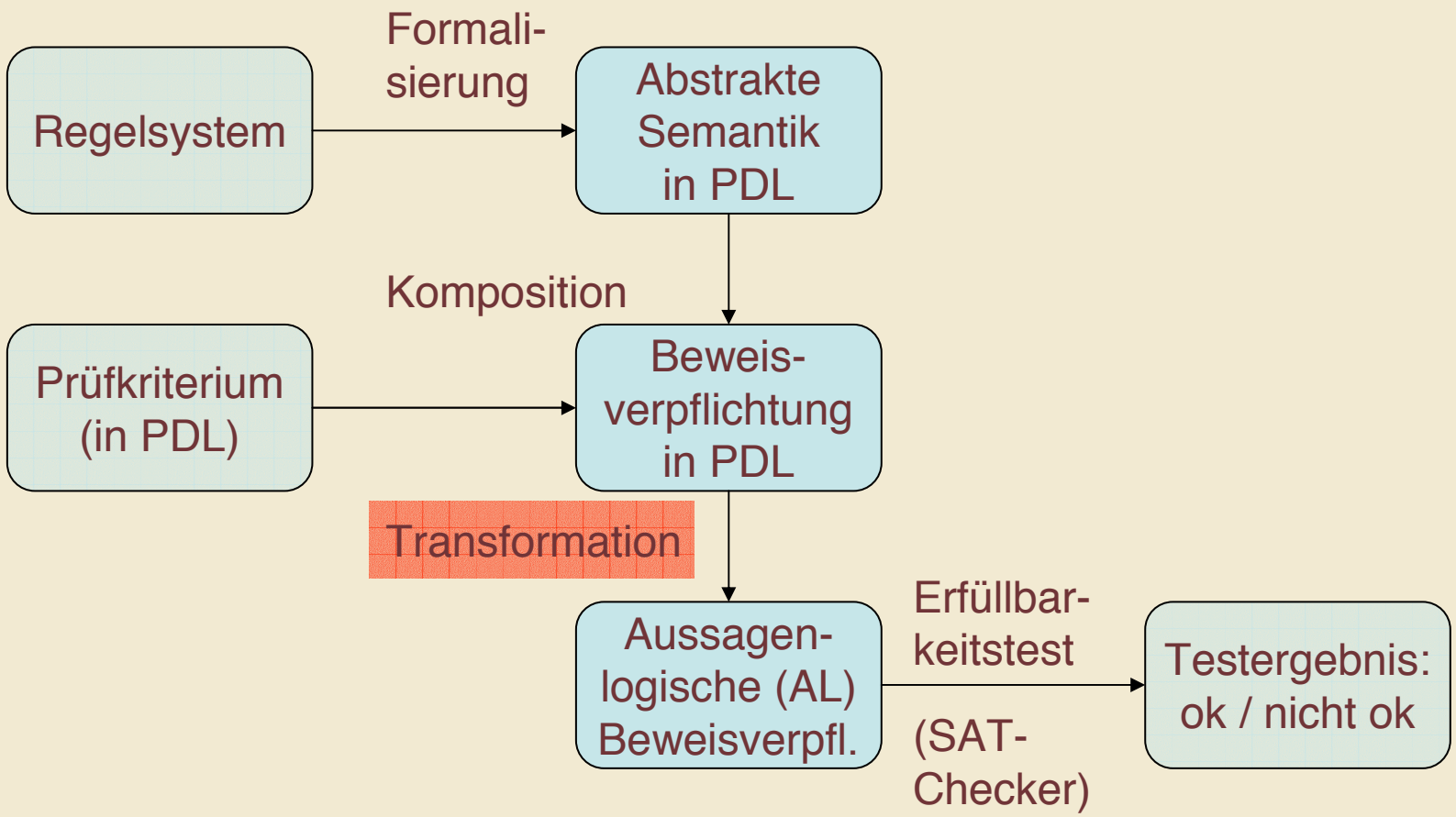
Konsistenz der Zusteuerung

Def.: Wenn es einen gültigen Auftrag gibt, der durch die Zusteuerung in einen nicht mehr gültigen transformiert wird, so heißt die Zusteuerung inkonsistent.

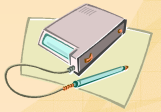
- Konsequenz: Aufträge werden unnötigerweise wegen fehlerhafter automatischer Modifikationen abgelehnt
- Formalisierung in PDL:

Zusteuerung konsistent gdw. $\langle B \rangle \mathsf{true} \Rightarrow [Z]\langle B \rangle \mathsf{true}$

Verifikationsmethode



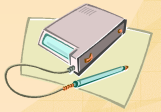
Transformation PDL nach AL



- Allgemeines Verfahren für *reguläre* PDL-Programme: **Termersetzungssystem**

$$
\begin{array}{lcl}
[x := b]F & \mapsto & F|_{x=b} \\
[\alpha \cup \beta]F & \mapsto & [\alpha]F \wedge [\beta]F \\
[\alpha ; \beta]F & \mapsto & [\alpha][\beta]F \\
[\alpha^*]F & \mapsto & \nu G.F \wedge [\alpha]G \\
[G?]F & \mapsto & G \Rightarrow F
\end{array}
$$

Spezialisierung Mercedes-Benz



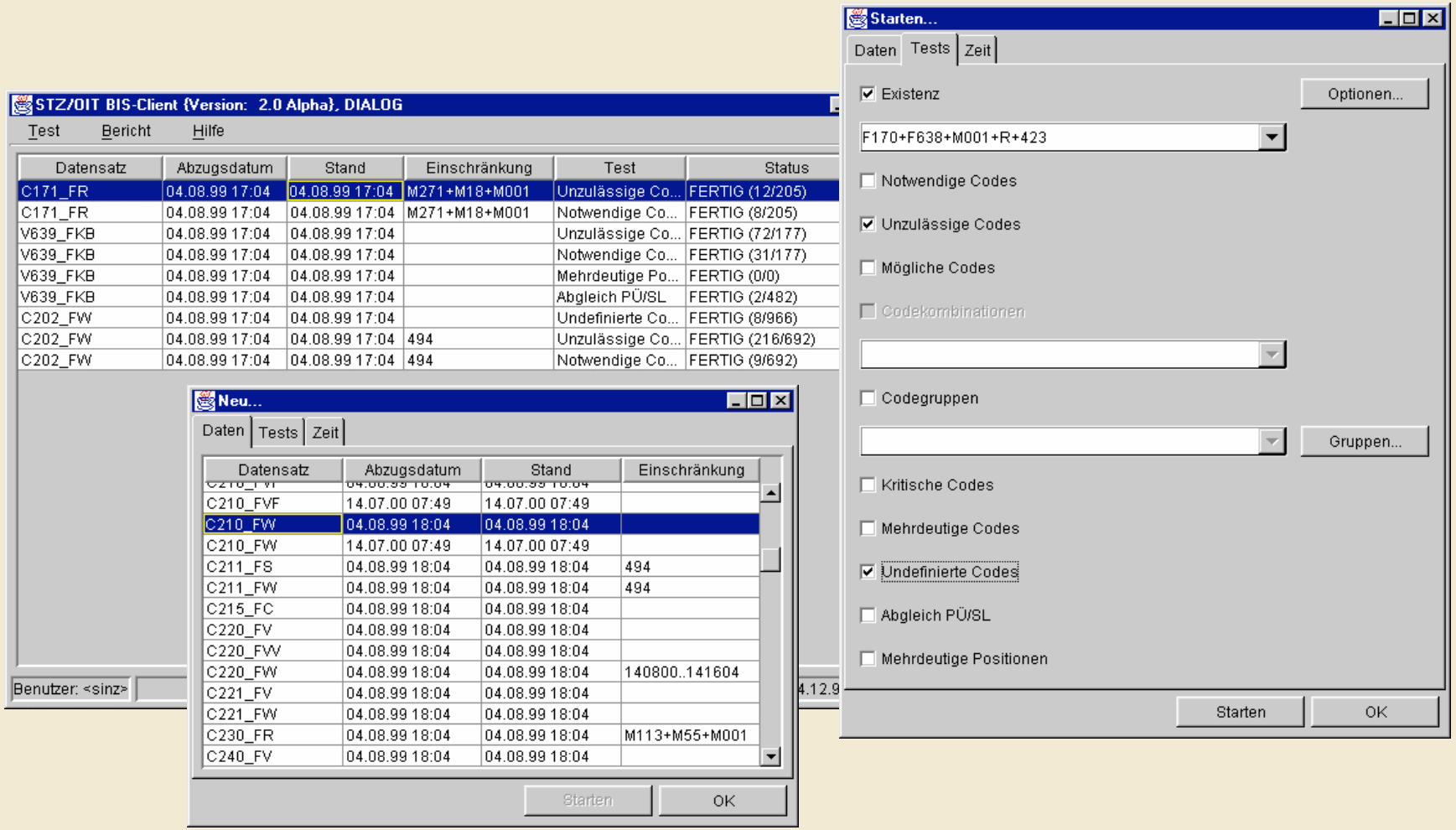
- Vermeidung der Fixpunkt-Berechnung (Ausnutzung der speziellen Programmstruktur)
- Rückführung fast aller Tests auf ein allgemeines aussagenlogisches Schema der Form $B \wedge E$, wobei

$$B := (Z_1 \Rightarrow c_1) \wedge \cdots \wedge (Z_n \Rightarrow c_n) \wedge (c_1 \Rightarrow B_1) \wedge \cdots \wedge (c_n \Rightarrow B_n)$$

die möglichen Zustände nach Z;B und vor T beschreibt.

Baubarkeits-Informations-System

>>> Verifikation Mercedes-Benz



Ergebnisse (Auszug)

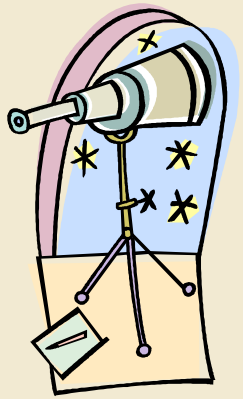
- Baureihe C202/FW, Stand 4. August 1999:
  - **M010 ist unzulässiger Code**

    M010 darf nur zusammen mit M111 und 801 verbaut werden, 801 ist aber nicht zulässig
  - **Zusteuerung ist inkonsistent**

    Zusteuerung von 956 führt bei Aufträgen, die 904U enthalten, zu einem ungültigen Auftrag
  - **Mehrere überflüssige Teile in der Stückliste**

    (bei insgesamt 18508 Stücklistenpositionen)

Teil 3: Empirische Feststellungen



Komplexität

- Aussagenlogische Beweise (SAT) auffallend einfach:
  - Probleme mit bis zu 1891 Variablen, 38113 Literalen
  - Bei 64328 Beweisen maximale Laufzeit von 0.148 sec. (maximale Suchraumgröße von 561 Knoten)
  - Moderne Methoden des SAT-Checking erforderlich
- Worin liegen die Ursachen?
  - Spezielle Struktur der Formel B

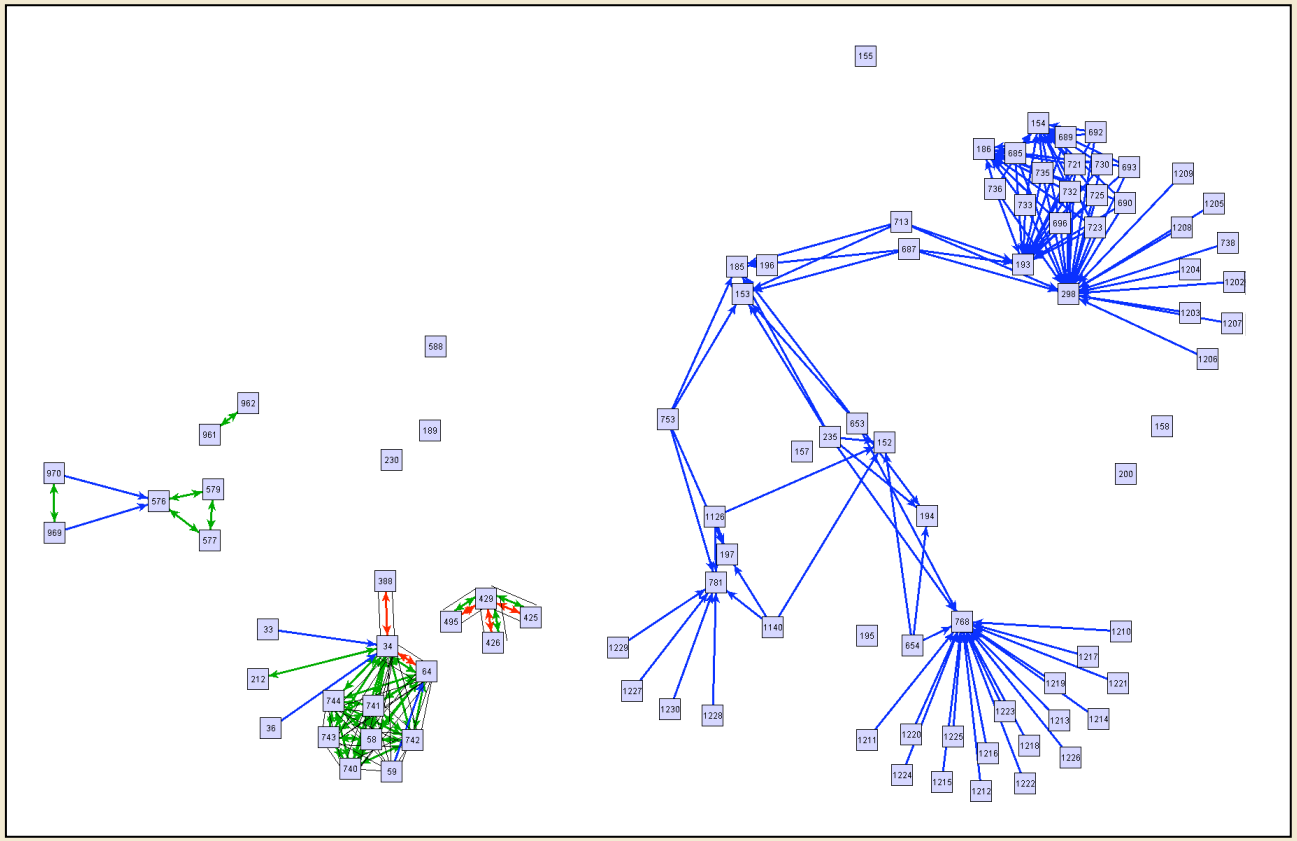
    (Jedoch keine bekannte polynomiell entscheidbare Teilklasse von SAT)
  - Natürliche Ordnung der Konfigurationsoptionen

Ordnung der Variablen

- „Wichtigkeit“ der Konfigurationsoptionen variiert
- Maßnahmen zur Nutzung:
  - Grundlegende, weitreichende Optionen (mit vielen Abhängigkeiten) zuerst betrachten
- Vorverarbeitung („Kompilierung“) der Formel B
  - Abschlussbildung unter geordneter Resolution
  - Einzelne Beweise der Form $B \wedge E$ dann durch erneute Abschlussbildung möglich („aktive Literale“)
  - Dadurch weitere Beschleunigung des Beweisprozesses

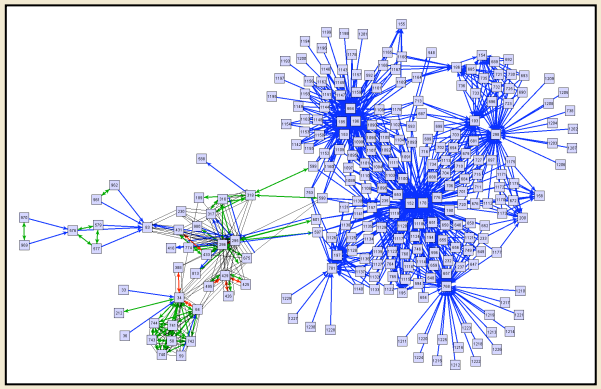
Innere Formelstruktur

Veranschaulicht durch (Variablen-)Interaktionsgraphen

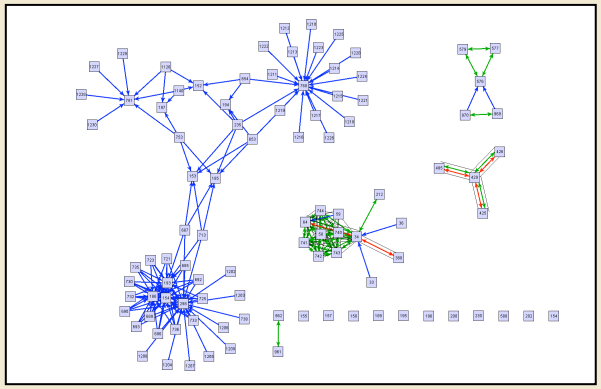


Interaktionsgraphen: Vergleich

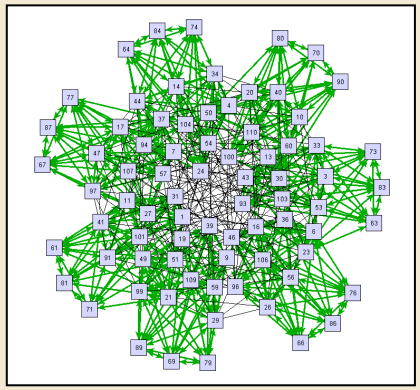
>>> Empirische Feststellungen



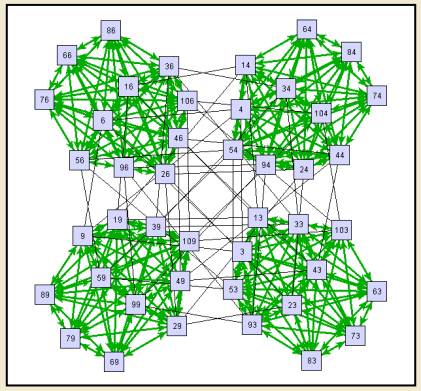
3 Schritte



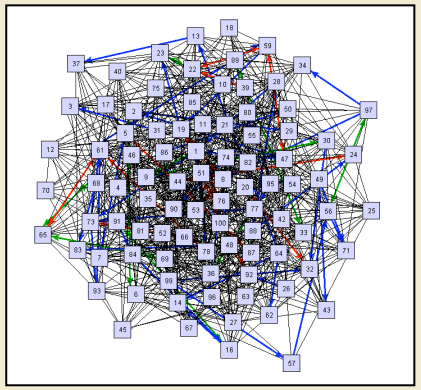
Konfiguration



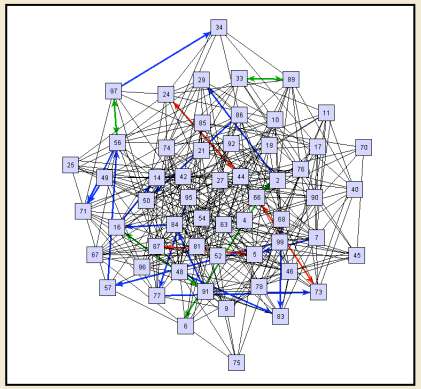
3 Schritte



Schubfach



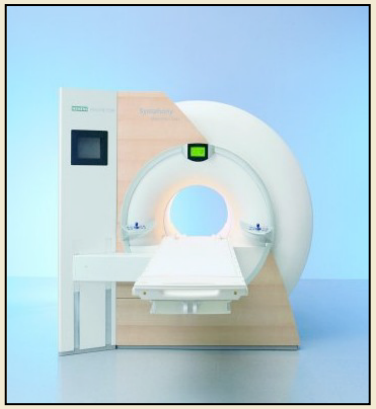
3 Schritte



Zufall

Weitere Arbeiten

- Konfiguration medizinischer Großgeräte
  - SIEMENS Magnetresonanztomografen
  - Vollständigkeitsprüfung der Online-Dokumentation



- Verifikation IBM Expertensystem
  - *IBM System Automation* (regelbasierte Steuerung von Rechner-Clustern)
  - Formalisierung in ΔPDL
  - Beweis von Terminationseigenschaften

Zusammenfassung

- Vorschlag einer formalen Semantik für regelbasierte Konfigurationsprogramme
- Erarbeitung eines Verifikationsverfahrens basierend auf einer Transformation von PDL nach AL
- Entwicklung eines Client/Server-Systems zur Prüfung der Mercedes-Benz-Produktdaten (BIS)
- Analyse und Nutzung der speziellen Formelstruktur durch Kompilierung und aktive Literale
- Entwicklung eines parallelen SAT-Checkers

Danksagung

Mein herzlicher Dank gilt

- Prof. Küchlin (Betreuung)
- allen Mitarbeitern des Arbeitsbereichs Symbolisches Rechnen, insbesondere
  - Wolfgang Blochinger (paralleles SAT-Checking)
  - Andreas Kaiser (Entwicklung BIS)
- dem Arbeitsbereich Paralleles Rechnen (*yfiles*)
- den Gutachtern meiner Dissertation

Schlussbetrachtungen

(Our work gives) strong evidence that it is not easy to determine whether a given proposition formula is a tautology, even if the formula is in normal disjunctive form.

S. Cook: *The Complexity of Theorem-Proving Procedures*. Proc. 3rd STOC (1971).

Our results suggest that the run time of search algorithms that exploit as much of the tractable structure as possible may in fact scale polynomially in the typical case.

R. Monasson et al.: *Determining computational complexity from characteristic 'phase transitions'*. Nature 400 (1999).