Fakultät Informatik Institut für Systemarchitektur, Professur Betriebssysteme

Schritte zur Portierung des Fiasco Mikrokerns auf PowerPC

Matthias Lange

Dresden, 5. Dezember 2006

Überblick

- Motivation
- Seitentabellenschnittstelle
- Implementierung
- Ergebnisse
- Zusammenfassung

Motivation

Motivation

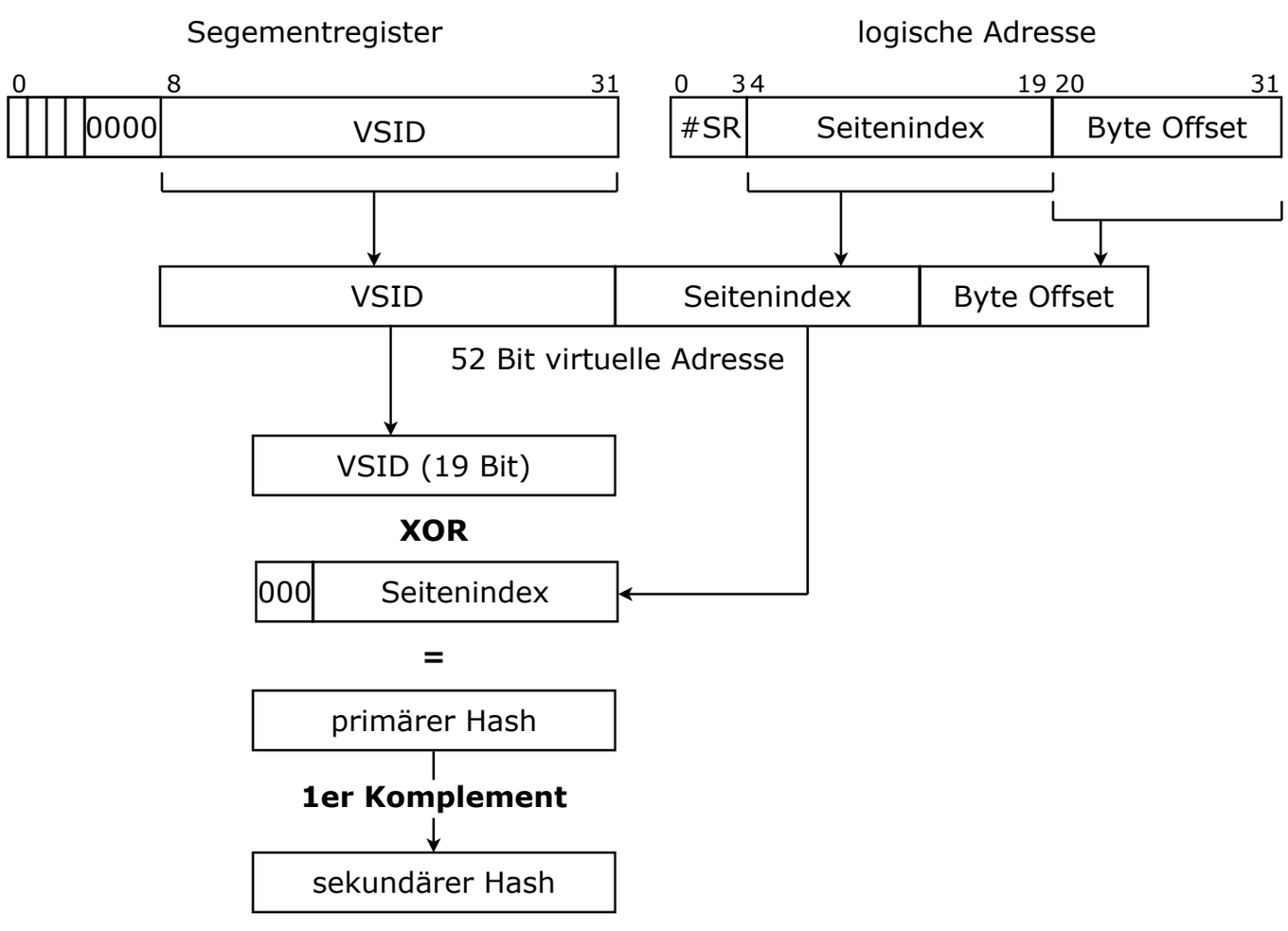
- flexible Konstruktion von Betriebssystemen mit Mikrokernen
- Fiasco auf unterschiedlichen Architekturen einsetzen
  - aktuell: IA-32, AMD64 und ARM
- strukturelle Verbesserungen im Kern
  - Wartbarkeit verbessern
  - Übertragbarkeit auf neue Plattformen (z.B. PowerPC) erleichtern

Seitentabellenschnitt-stelle

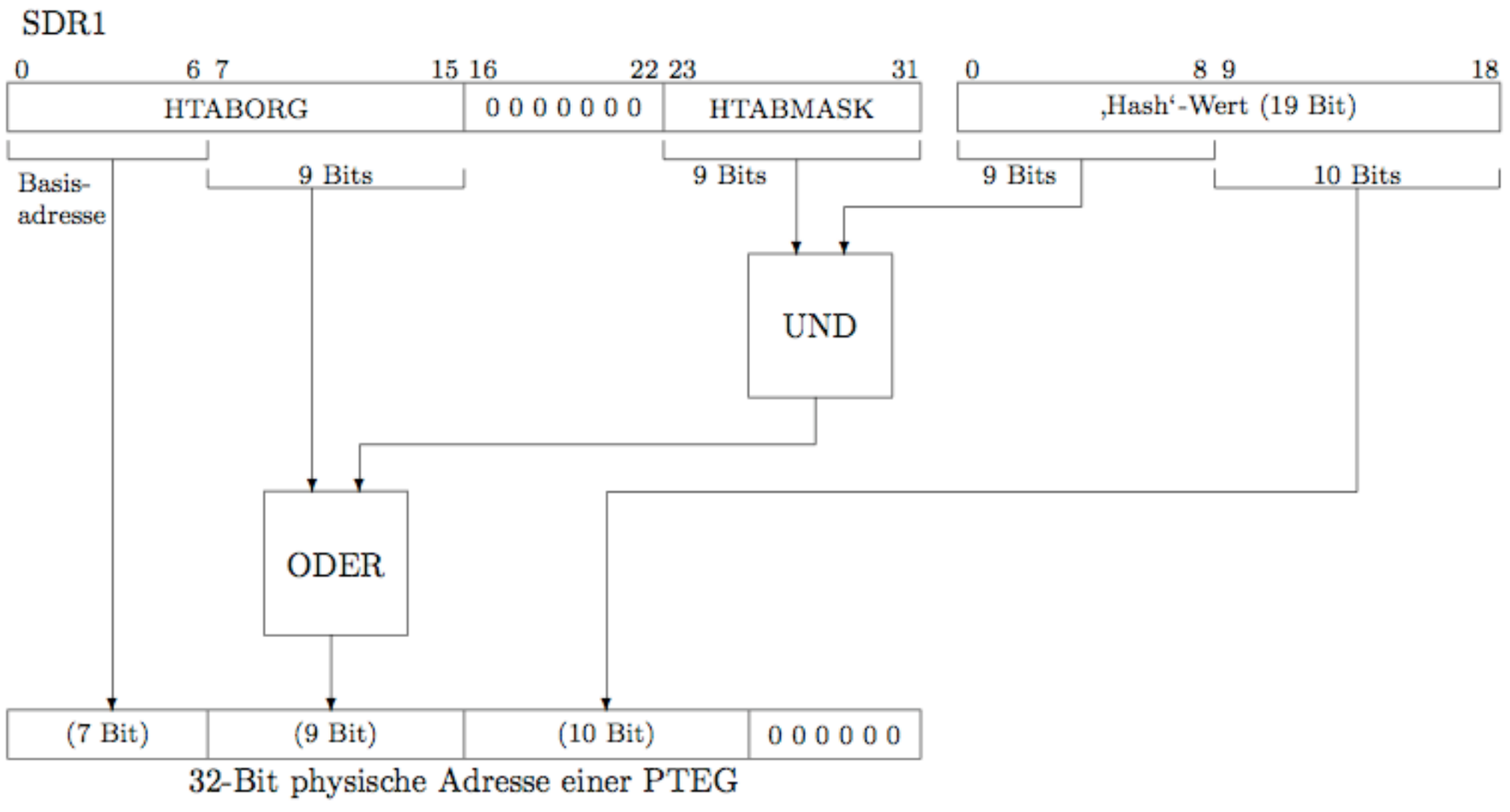
PPC Adressumsetzung

- Adressraum in 16 Segmente zu je 256 MB eingeteilt
  - Beschreibung durch Segmentdeskriptoren (VSID)
- Seitengröße ist 4 KB
- Verwendung von hashed-Seitentabellen
  - eine Seitentabelle für alle Adressräume
  - Seitentabelle besteht aus Gruppen (PTEG) zu je acht Einträgen
  - Hashfunktion wählt eine Gruppe aus, deren Einträge mit der virtuellen Adresse verglichen werden

PPC Adressumsetzung



PPC Adressumsetzung



32-Bit physische Adresse einer PTEG

Adressumsetzung

- Einträge der PTEG werden mit der virtuellen Adresse verglichen
  - Eintrag in der primären oder sekundären PTEG
  - gültiger Eintrag
  - Übereinstimmung des VSID-Felds
  - evtl. werden weitere Felder verglichen (z.B. bei mehreren Matchings)

Seitentabellenschnittstelle

- architekturneutrale Abstraktion der MMU-Hardware
- Ziele
  - Unterstützung mehrere Seitengrößen
  - Unterstützung beliebigstufiger Seitentabellen
  - Unterstützung für hashed-Seitentabellen
  - alle Modifikationen der Seitentabelle unter dieser Schnittstelle zusammenfassen (Struktur und Wartbarkeit des Kerns verbessern)

Ansätze

- Mach
  - arbeitet mit Speicherobjekten (pmap-Ebene), die den darunter liegenden physischen Speicher verwalten
  - Funktionen zum Erzeugen, Einfügen, Entfernen und Ändern von Speicherobjekten
- Linux
  - 3-stufige Seitentabelle für jeden Adressraum (seit kurzem 4 Stufen)
  - “Zusammenfalten” bei Architekturen mit weniger Stufen

Ansätze

- Fiasco AMD64
  - Erweiterung des IA-32-”Systems” für 4-stufige Seitentabellen
    - weiterhin direkte Manipulation der Seitentabelle
  - geringfügige strukturelle Verbesserungen (z.B. map_superpage)
- Fiasco ARM
  - Separation in architekturspezifische und -neutrale Teile
  - semantisch klar definierte Funktionen (insert, lookup, ...)

Spezielle Anforderungen

- Fiasco
  - geht von hierarchischen Seitentabellen aus
  - Einträge in der Mapping-Datenbank sind immer über die Seitentabelle zugreifbar
- PowerPC
  - eine Seitentabelle für alle Adressräume (hashed-Seitentabelle)
  - Einträge können aus der Seitentabelle verdrängt werden
- Fiasco-Annahmen sind für PPC nicht erfüllt

Lösungsmöglichkeiten

- Einführung von kerninternen Speicherobjekten, die in einer eigenen Struktur verwaltet werden
  - grundlegende Änderungen an den Kernsystemen (Mapping-Datenbank, Infrastruktur für die Verwaltung der Objekte, Anpassungen aller Systeme an die neue Semantik im Kern)
- Semantik von Speicherseiten beibehalten
  - zusätzliche hierarchische Seitentabelle für jeden Adressraum
  - diese speichert alle oder nur verdrängte Umsetzungen

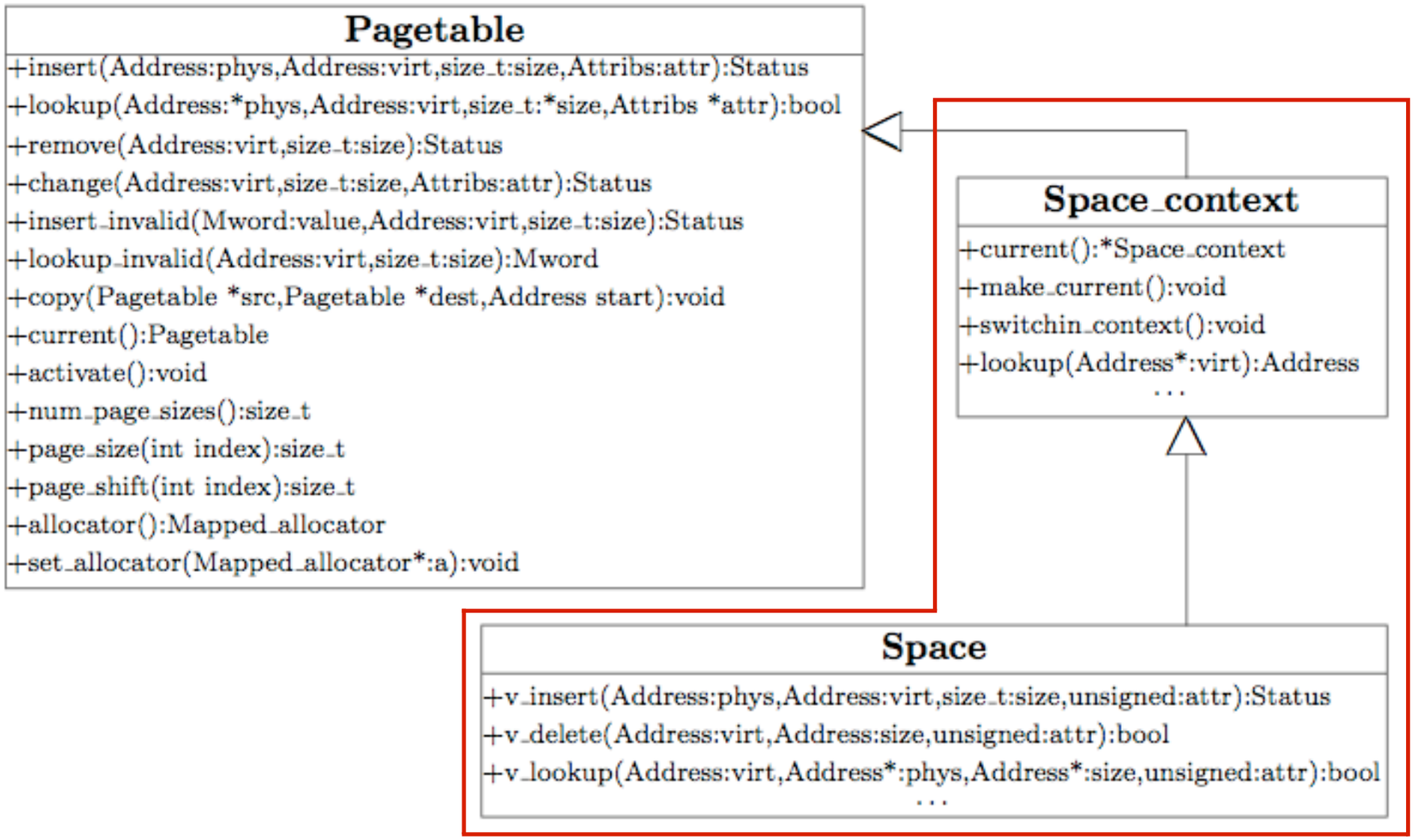
Entwurf

- Klasse Pagetable wird als neue Abstraktions-schicht eingeführt
- Funktionen zum Einfügen, Löschen, Suchen, Ändern und Kopieren von Seitentabelleneinträgen
- implizites Wissen über Seitengrößen wird durch Funktionen explizit in der Schnittstelle ausgedrückt

Entwurf

- Fiasco-spezifische Funktionen zum Einfügen und Suchen ungültiger Seiten
- Konstantennamen für Seitenattribute, die im architekturspezifischen Teil festgelegt werden
  - festgelegte Semantik

Entwurf


Pagetable
+insert(Address:phys,Address:virt,size_t:size,Attribs:attr):Status
+lookup(Address:*phys,Address:virt,size_t:*size,Attribs *attr):bool
+remove(Address:virt,size_t:size):Status
+change(Address:virt,size_t:size,Attribs:attr):Status
+insert_invalid(Mword:value,Address:virt,size_t:size):Status
+lookup_invalid(Address:virt,size_t:size):Mword
+copy(Pagetable *src,Pagetable *dest,Address start):void
+current():Pagetable
+activate():void
+num_page_sizes():size_t
+page_size(int index):size_t
+page_shift(int index):size_t
+allocator():Mapped_allocator
+set_allocator(Mapped_allocator*:a):void
Space_context
+current():*Space_context
+make_current():void
+switchin_context():void
+lookup(Address*:virt):Address
...
Space
+v_insert(Address:phys,Address:virt,size_t:size,unsigned:attr):Status
+v_delete(Address:virt,Address:size,unsigned:attr):bool
+v_lookup(Address:virt,Address*:phys,Address*:size,unsigned:attr):bool
...

Implementierung

Seitentabellenschnittstelle

- für Fiasco-V4 für IA-32 implementiert
- Seitentabellenrepräsentation wurde teilweise von der IA-32 Implementierung übernommen
- Beachtung der unterschiedlichen Semantik von Einfüge-Operationen
  - Einfügen von Umsetzungen (map)
  - Kopieren von Seitentabellenverzeichniseinträgen (Sharing)
  - Einfügen ungültiger Seiten

PowerPC

- Skizzierung der Implementierung
- Seitentabellenrepräsentation: Feld von PTEGs aus je 8 PTEs
- zusätzliche Seitentabelle als zweistufige Seitentabelle implementieren
- Wahl der VSID: physische Adresse des Pagetable-Objekts

Ergebnisse

Ergebnisse

- strukturelle Verbesserungen
  - keine direkte Seitentabellenmodifikation im Kern mehr
  - Komplexität des Quellcodes ist um 170 Zeilen (0,4%) gesunken
- Architekturneutralität
  - Definition von Funktionen, die ein möglichst breites Spektrum an Anwendungsfeldern abdecken
  - Größenparameter ermöglicht Unterstützung mehrstufiger Seitentabellen und unterschiedlicher Seitengrößen

Messungen

- Pingpong- und Mikrobenchmark auf drei unterschiedlichen Testrechnern

Prozessor	**Speicher**
Pentium III, 450MHz	256MB
AMD Sempron, 1,5GHz	256MB
Pentium IV, 2,8GHz	256MB

Mikrobenchmark

- kerninterne Sequenz von v_lookup, v_insert, v_lookup und v_delete Aufrufen
- Messung Taktzyklen für 500.000 Durchläufe
- Overhead zwischen 19% (AMD) und 26% (P3)
  - erweitertes insert um lookup zu sparen?

	v4	v4 mit Interface	Δ (in %)
AMD	71.735.393	85.270.588	19
P III	77.203.330	97.606.104	26
P IV	93.002.784	112.453.663	20

Pingpong

- Inter-AS IPC-Benchmark
- lookup beim Kontextwechsel
- kaum Overhead (0% bis 1%)
  - andere Kosten überwiegen (Kernein- und -austritte)

	v4	v4 mit Interface	Δ (in %)
AMD	1192	1185	0,5
P III	517	512	1
P IV	7721	7720	0

Zusammenfassung

Zusammenfassung

- Entwicklung einer architekturneutralen Seitentabellenschnittstelle
  - Ziele
  - Betrachtungen von Mach, Linux, Fiasco AMD64 und ARM
  - besondere Problematik mit Fiasco-Annahmen und hashed-Seitentabellen (PowerPC)
  - Entwurf der Seitentabellenschnittstelle

Zusammenfassung

- Implementierung
  - Implementierung der Schnittstelle für IA-32
  - Ausblick für die Implementierung für PowerPC
- Ergebnisse
  - strukturelle Verbesserungen wurden erreicht
  - Architekturneutralität ist gewährleistet
  - bis zu 1% Overhead für Userland
  - kernintern bis zu 26% Overhead, durch Verbreiterung der Semantik der insert-Funktion evtl. vermeidbar

Fragen & Diskussion