

SS 16

Luke Hain

12. April 2016

# Inhaltsverzeichnis

|            |   |           |
|------------|---|-----------|
| <b>I</b>   | <b>Computer Networks</b>                | <b>3</b>  |
| <b>1</b>   | <b>Vorlesung</b>                        | <b>4</b>  |
| 1.1        | Einführung . . . . .                    | 4         |
| 1.2        | Bitübertragungsschicht . . . . .        | 5         |
| <b>2</b>   | <b>Übung</b>                            | <b>6</b>  |
| 2.1        | Einführung . . . . .                    | 6         |
| 2.1.1      | . . . . .                               | 6         |
| 2.1.2      | . . . . .                               | 7         |
| 2.1.3      | . . . . .                               | 7         |
| 2.1.4      | . . . . .                               | 8         |
| <b>II</b>  | <b>Theoretical Informatic and Logic</b> | <b>10</b> |
| <b>3</b>   | <b>Vorlesung</b>                        | <b>11</b> |
| 3.1        | Prädikatenlogik erster Stufe . . . . .  | 11        |
| 3.2        | Prädikatenlogik erster Stufe . . . . .  | 12        |
| <b>4</b>   | <b>Übung</b>                            | <b>14</b> |
| 4.1        | Prädikatenlogik . . . . .               | 14        |
| <b>III</b> | <b>Computer Architecture</b>            | <b>15</b> |
| <b>5</b>   | <b>Vorlesung</b>                        | <b>16</b> |
| 5.1        | Einführung . . . . .                    | 16        |
| 5.1.1      | Big Data . . . . .                      | 16        |
| <b>6</b>   | <b>Übung</b>                            | <b>17</b> |
| 6.1        | Einführung . . . . .                    | 17        |

|           |                            |           |
|-----------|----------------------------|-----------|
| <b>IV</b> | <b>Database</b>            | <b>18</b> |
| <b>7</b>  | <b>Vorlesung</b>           | <b>19</b> |
| 7.1       | Einführung . . . . .       | 19        |
| <b>8</b>  | <b>Übung</b>               | <b>20</b> |
| 8.1       | Einführung . . . . .       | 20        |
| <b>V</b>  | <b>Hardware Internship</b> | <b>21</b> |
| <b>VI</b> | <b>C++4CG</b>              | <b>22</b> |

# Teil I

## Computer Networks

# Kapitel 1

## Vorlesung

### 1.1 Einführung

- Anwendungsfelder Rechnernetze (1.4)
  - Geschäftsanwendungen - gemeinsame Nutzung von Ressourcen
  - Privatbereich - Informationszugriff (z.B. WWW, IM)
  - Mobile Benutzer - Textnachrichten, ...
  - Gesellschaftliche Aspekte - Copyright, Profile, ...
- Client Server Modell (1.5)
- Peer-to-Peer Communication (1.6)
- Basis-Netzstruktur (1.7)
  - Übertragungsmodi
    - \* Verbindungsorientiert
    - \* Verbindungslos (z.B. IP)
    - \* Leitungsvermittelt
    - \* Paketvermittelt (flexibler, ressourcenschonend)
- Schichtenarchitektur - ISO/OSI Referenzmodell (1.8)
  - International Organization for Standardization
  - Open Systems Interconnection
  - Schichtenübersicht auf 1.8 ff.
- Integriertes Referenzmodell (Tanenbaum) (1.11)
  - Protokollimplementierung oft abweichend vom Referenzmodell

- Beispiel Datenübertragung (1.12)
- Schichteneffizienz (1.13)
- Dienste - Begriffsklärung (1.14)
  - Beispiel Ablaufdiagramm (1.15)
- Netzkopplung - Basis-Topologien
  - Punkt-zu-Punkt-Kanäle (Unicast)
  - Rundsendekanäle (Broadcast)
  - Klassifizierung nach Ausdehnung (1.17)
    - \* Pan - Personal Area Network
    - \* LAN - Local Area Network
    - \* MAN - Metropolitan Area Network
    - \* WAN - Wide Area Network (1.18)
  - Mobilität || Leistung (1.19)
  - Konzepte - Layer-N-Gateway(1.20)
  - Beispiel (1.21)
- Internet(1.22 ff)
  - Internet
    - \* Geschichte des Internet (1.24 ff)
    - \* Normen (1.26)
  - Intranet (1.22)

## 1.2 Bitübertragungsschicht

# Kapitel 2

## Übung

### 2.1 Einführung

timo.schick@tu-dresden.de

#### 2.1.1

- a) Sterntopologie: Ein zentrales Element (Sternkoppler), jeder Rechner benötigt eine Leitung zu Sternkoppler  $\rightarrow 5$
- b) Jeder mit Jedem  $= 4 + 3 + 2 + 1 = 10$
- c) (1)  $l(n) = n$  bei Sterntopologie  
(2)  $l(n) = \sum \dots = (n * (n - 1)) / 2$  bei vollvermaschter Topologie
- d) (1) LAN
  - Reichweite: 10m
  - Reaktionszeit: niedrig
  - Datenrate: hoch
  - Topologien: Sterntopologie
- (2) MAN
  - Reichweite: 10km
  - Reaktionszeit: mittel
  - Datenrate: mittel
  - Topologien: hierarchische Topologie
- (3) WAN
  - Reichweite: 100km - 10.000km
  - Reaktionszeit: hoch
  - Datenrate: niedrig
  - Topologien: Vollvermaschte Topologie

### 2.1.2

a) Dienst und Protokoll

- siehe Musterlösung

b) OSI Schichtenmodell

- Schichtenmodell siehe Folie 1.8ff
- Protokoll:
  - ist eine Sprache zur horizontalen Kommunikation zwischen Prozessen derselben Schicht auf verschiedenen Hosts
- Dienst
  - dient der vertikalen Kommunikation zwischen zwei Schichten auf einem Host
- Aufteilung des Bitstroms: Schicht 2 Sicherungsschicht
- Ende-zu-Ende Kommunikation: Schicht 4 Transportschicht
- Wegewahl: Schicht 3 Vermittlungsschicht

c) keine inhaltliche Bearbeitung, sondern nur Informationsweiterleitung

### 2.1.3

- a)
- siehe Folie 1.15;
  - Initiator (Prozess A), ...
  - Responder (Prozess B), ...

b) (1) Zustände bestimmen

- idle
- connected
- prepare(Initiator)
- prepare(Responder)

(2) Übergänge bestimmen (Knoten, Pfad, Knoten)

- (idle, conReq, prep(Init))
- (idle, ConInd, prep(Resp))
- (prep(Resp), conRsp, connected)
- (prep(Init), conCnf, connected)
- (connected, dataRep/dataInd, connected)
- (prep(Resp)/prep(Init)/connected, disRep/disInd, idle)

c) (1) Ablaufdiagramm



- c1) + zeitlicher Ablauf
- c2) - es werden n Diagramme benötigt
- c3) -

(2) Zustandsdiagramm

- c1) -
- c2) + alle Abläufe in einem Diagramm darstellbar
- c3) +

#### 2.1.4

a) siehe Folie 1.10

$$(1) \ PDU(N) = SDU(N - 1)$$

$$(2) \ IDU(N) = ICI(N) + SDU(N)$$

b) Seitenaufruf: <http://www.heise.de/software>

(1) httpRequest

- i. GET/software/http/1.1
- ii. Host: www.heise.de

(2) ICI

- i. ip: 193.99.144.85 port:80

(3) SDU

- i. GET/software/http/1.1
- ii. Host: www.heise.de

(4) IDU

- i. ICI
- ii. SDU

(5) TCP-PDU

- i. src:80, dest:80,...
- ii. SDU
- iii. Data

c)

$$b_0 = 125 \frac{\text{Mbit}}{\text{s}}$$

$$b_1 = b_0 \cdot 0,8$$

$$b_2 = b_1 \frac{(55 + 99)0,01}{2}$$

$$b_3 = b_2 \frac{(57 + 99)0,01}{2}$$

$$b_4 = b_3 \frac{(23 + 99)0,01}{2} = 36,4 \frac{\text{Mbit}}{\text{s}}$$

$$b_4 = b_{\text{goodput}}$$

$$b_{\text{extra}} = b_2 \frac{(23 + 99)0,01}{2} = 46,7 \frac{\text{Mbit}}{\text{s}}$$

## Teil II

# Theoretical Informatic and Logic

# Kapitel 3

## Vorlesung

### 3.1 Prädikatenlogik erster Stufe

- Syntax
  - Ein Alphabet der Prädikatenlogik besteht aus ... (2)
  - forall heist universeller Quantor, exists heißt existenzieller Quantor
  - Funktions- und Relationssymbolen ist eine Stelligkeit  $n \in \mathbb{N}$
  - Nullstellige Funktionssymbole werden als ... (3)
- Terme
  - Definition 4.2 prädikatenlogische Terme (4)
  - Ein Term ist abgeschlossen oder grundinstanziiert, wenn in ihm keine Variablen vorkommen
  - Die Menge der abgeschlossenen Terme wird mit  $T(F)$  bezeichnet
- Prädikatenlogische Atome (5)
- Prädikatenlogische Formeln (6)
  - prädikatenlogische Formeln
- Strukturelle Rekursion
  - Rekursionssätze lassen sich für  $T(F, V)$  und  $L(R, F, V)$  formulieren
  - Es gibt genau eine Funktion  $foo$  die die folgenden Bedingungen erfüllt: (7)
    - \* Rekursionsanfang
    - \* Rekursionsschritt
  - Beispiele (8/9)

## 3.2 Prädikatenlogik erster Stufe

- Strukturelle Induktion
  - Induktionssätze lassen sich für  $T(F, V)$  und  $L(R, F, V)$  formulieren
  - jeder Term besitzt die Eigenschaft E, wenn: (10)
  - analog für prädikatenlogische Formeln
- Aufgabe (11)
  - Beweisen Sie, dass  $\forall F \in L(R, F, V)$  die Aussage  $l'(m(F)) \geq l(F)$  gilt
- Teilterme und Teilformeln (12)
  - Die Def. 3.8 lässt sich auf Terme und Formeln übertragen
  - Beispiel
- Freie und gebundene Vorkommen einer Variablen (13)
  - Def. 4.5 Die **freien Vorkommen einer Variablen** in einer prädikatenlogischen Formel sind wie folgt definiert: (13)
- Abgeschlossene Terme und Formeln (14)
  - nach Def. 4.2: Ein abgeschlossener Term ist ein Term, in dem keine Variable vorkommt
  - Def. 4.6 Eine abgeschlossene Formel (oder kurz ein Satz) der Sprache  $L(R, F, V)$  ist eine Formel der Sprache  $L(R, F, V)$ , in der jedes Vorkommen einer Variablen gebunden ist
- Substitutionen (19)
  - Def. 4.7: Eine **Substitution** ist eine Abbildung  $\sigma : V \rightarrow T(F, V)$ , die bis auf endlich viele Stellen mit der Identitätsabbildung übereinstimmt
  - Beispiel
- Instanzen
  - Statt  $\sigma(X)$  schreiben wir in der Folge  $X\sigma$
  - Def. 4.8: Sei  $\sigma$  eine Substitution  $\sigma : V \rightarrow T(F, V)$  kann wie folgt zu einer Abbildung  $\sigma_{dach} : T(F, V) \rightarrow T(F, V)$  erweitert werden: (25)
  - Grundinstanz
  - Proposition
- Komposition von Substitutionen

- Def. 4.10: Seien  $\sigma$  und  $\theta$  zwei Substitutionen Die Komposition  $\sigma\theta$  von  $\sigma$  und  $\theta$  ist die Substitution: (30)
  - Aufgaben
- Komposition von Substitutionen (33)

# Kapitel 4

## Übung

### 4.1 Prädikatenlogik

# Teil III

## Computer Architecture



# Kapitel 5

## Vorlesung

### 5.1 Einführung

#### 5.1.1 Big Data

"Big Data hat die Chance die geistige Mittelschicht in Hartz IV zu bringen"

# Kapitel 6

## Übung

### 6.1 Einführung

# Teil IV

## Database

# Kapitel 7

## Vorlesung

### 7.1 Einführung

Gründe für DBS-Einsatz:

- Effizienz und Skalierbarkeit
- Fehlerbehandlung und Fehlertoleranz
- Mehrbenutzersynchronisation

ANSI - Database

- Standard siehe 1VL

Geschichte der Datenbanktechnologie

- siehe 1VL(28 ff.)

Databases vs Information Retrieval

- Information Retrieval 1VL(44)
  - Suche nach Dokumenten
  - Nimmt ständig zu
  - In welchem Datenbestand wird gesucht? etc...

Databases vs Big Data

- Big Data 1VL(47)

# Kapitel 8

## Übung

### 8.1 Einführung

# Teil V

## Hardware Internship

Teil VI

C++4CG