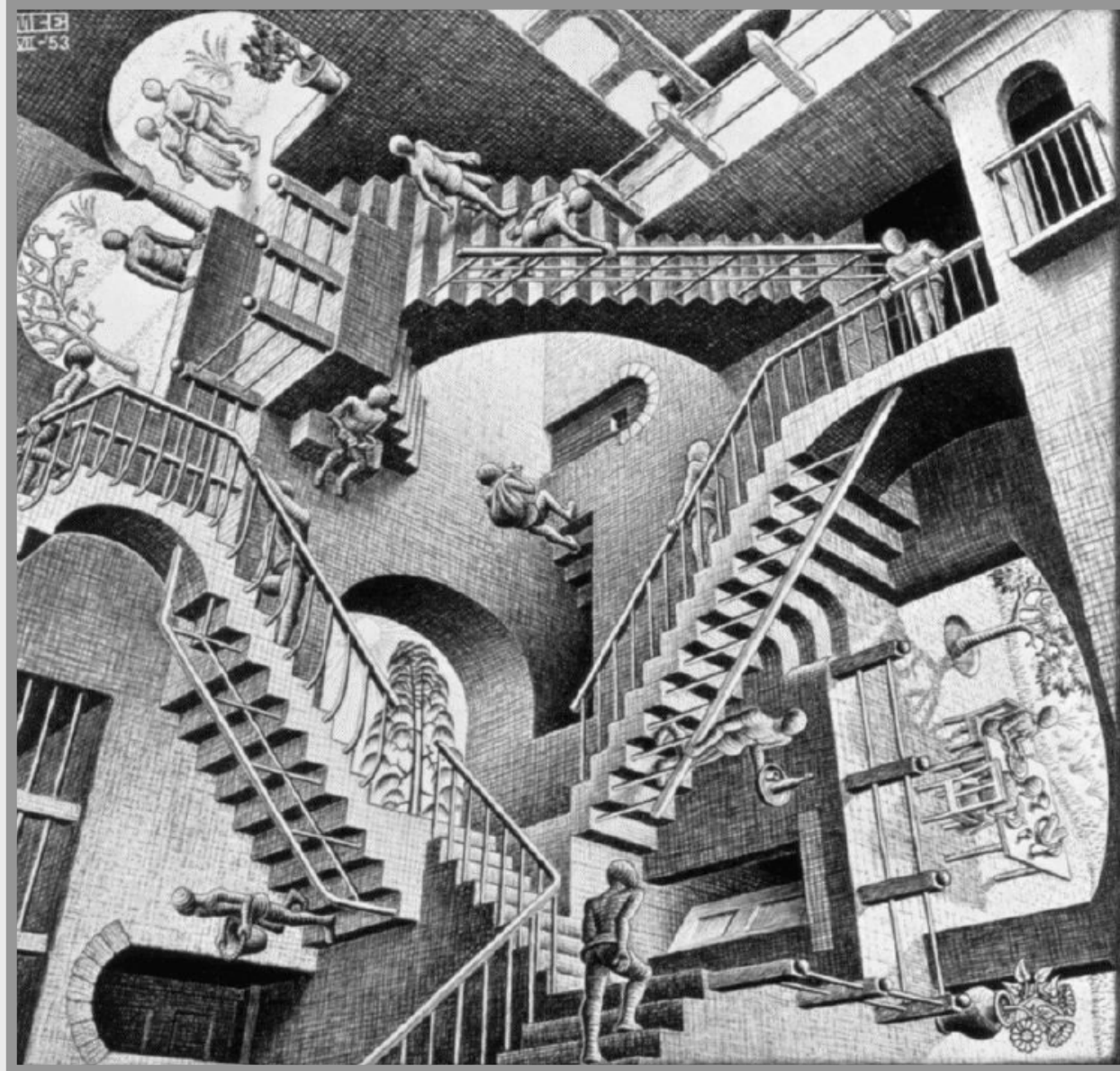


Wissens- basierte Diagnostik

Kap. 7:
Case Based Reasoning Teil 2

Dr. Norbert Waleschkowski
Fachbereich Informatik

Vorlesung Master-Studiengang
Wintersemester 2009/10



Ähnlichkeitsmaße (1)

- Ähnlichkeitsmaße dienen zur Bestimmung der Ähnlichkeit von neuen Diagnoseproblemen zu bekannten Fällen. (Anmerkung: Der Begriff der Ähnlichkeit kann verallgemeinert und auf die Ähnlichkeit von Objekten einer Menge O übertragen werden.)
- Im engeren Sinne benötigt man allerdings nur eine Präferenzordnung über den Fällen der Fallbasis.
- Ein Ähnlichkeitsmaß bildet jeden Fall auf einen numerischen Wert ab. Die Ordnung über den reellen Zahlen induziert dann die Präferenzordnung über der Fallbasis.
- Typischerweise wird als Bildraum das Intervall $[0, 1]$ gewählt. Dabei bedeuten „0“ keine (bzw. minimale) und „1“ maximale Ähnlichkeit.

Ähnlichkeitsmaße (2)

- Zunächst eine Definition: Sei CASE die Menge aller bekannten Fälle. Sei C die Menge aller Probleme, die man fallbasiert lösen kann. C ist also die Menge aller bekannten Fälle sowie aller Problemstellungen, die sich potentiell fallbasiert lösen lassen. Es gilt daher: $CASE \subset C$.
- Wir formulieren nun Axiome, die für Ähnlichkeitsmaße gelten sollten. Die Anforderungen an solche Maße, die wir im folgenden *sim* nennen, sind denkbar gering.

Zunächst gelte $sim: C \times C \rightarrow [0, 1]$.

- Axiom 1 (Reflexivität): Ein Objekt $c \in C$ (Fall bzw. Problem) ist sich selbst maximal ähnlich, d.h.

$$sim(c, c) = 1 \quad \forall c \in C$$
- Axiom 2 (Symmetrie): Zwei Objekte sind Ähnlichkeitssymmetrisch:

$$sim(c_1, c_2) = sim(c_2, c_1) \quad \forall c \in C$$

Die Gültigkeit dieses Axioms wird manchmal hinterfragt. So kann etwa diese Eigenschaft bei diagnostischen Aufgabenstellungen wegen der Beschränktheit der Information nicht immer gewährleistet werden.

Alternativ könnte auch ein Abstandsmaß geführt werden.

Definition: Eine Abbildung

$d: C \times C \rightarrow \mathcal{R}$ mit
 $d(x, x) = 0 \quad \forall x \in C$
 heißt Abstandsmaß.

Die Ermittlung der Ähnlichkeit ist ein dynamischer Prozeß. Zu Beginn einer Diagnose-sitzung sind i.d.R. erst wenige Parameter bekannt. Typischerweise verändern sich die Ähnlichkeitswerte während einer Sitzung.

Ähnlichkeitsmaße (3)

- Klassiker und Urvater der allermeisten Maße ist das gewichtete Hamming-Maß. Es arbeitet auf Attribut-Wert-Darstellungen mit booleschen Werten (boolesche Vektoren). Es ist wie folgt definiert:

- Def.: Ein gewichtetes Hamming-Maß H_g ist erklärt durch

$$H_g((a_1, \dots, a_n), (b_1, \dots, b_n)) = \sum_{i=1, \dots, n} (g_i \mid a_i = b_i, 1 \leq i \leq n)$$

wobei $g = (g_1, \dots, g_n)$, $0 \leq g_i \leq 1$, ein Gewichtsvektor ist. Oft wird g normiert, d.h. dass $\sum_i g_i = 1$.

- Gewichtete Hamming-Maße haben ihre Grenzen. Betrachte z.B. das XOR-Problem. XOR teilt die booleschen Vektoren $(0,0)$, $(0,1)$, $(1,0)$ und $(1,1)$ in zwei Klassen: $K_1 = \{(0,1), (1,0)\}$ und $K_2 = \{(0,0), (1,1)\}$
- Kein gewichtetes Hamming-Maß ist in der Lage, bei drei gegebenen Vektoren den jeweils vierten korrekt zu klassifizieren. So ist z.B. der Vektor $(0,0)$ dem Vektor $(0,1)$ ähnlicher als dem Vektor $(1,1)$, wird also der falschen Klasse zugeordnet. Der Grund liegt darin, dass die XOR-Funktion keine Ähnlichkeitsoperation durchführt. Das Problem ist bekannt aus der Neuroinformatik: Ein Perzeptron kann das XOR-Problem nicht lösen.

Ähnlichkeitsmaße (4)

- Dieses Problem kann durch Einführung lokaler Maße gelöst werden.
- Dabei basiert die Bestimmung der Ähnlichkeit von zwei Fällen auf der Betrachtung der relevanten lokalen Elemente der beiden Fälle. Man unterscheidet also zwischen lokaler Ähnlichkeit auf der Ebene der lokalen Elemente und der globalen Ähnlichkeit auf Fallebene.
- Das globale Ähnlichkeitsmaß sim wird verstanden als ein Maßvektor $sim = (sim_1, \dots, sim_n)$, wobei sim_i die Ähnlichkeit der Werte des i -ten Attributs ist (lokale Ähnlichkeit).

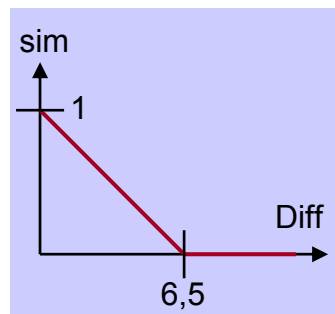
Batteriespannung U:

$ U_a - U_b $	$sim(U_a, U_b)$
[0,0-0,2)	1
[0,2-0,9)	0,9
[0,9-1,5)	0,85
[1,5-2,5)	0,7
[2,5-3,5)	0,5
[3-5-4,5)	0,2
[4,5-6,5]	0,1
>6,5	0

- Bsp.: Wir betrachten das Attribut Batteriespannung für zwei Fälle F_a und F_b . Die lokale Ähnlichkeit der Spannungswerte wird über die Spannungsdifferenz $|U_a - U_b|$ ermittelt (s. Tab. links u. Bsp. S.9).
- Das gewichtete globale Hammingmaß sei dann wie folgt definiert: Betrachte zwei Fälle a und b . Sei $g = (g_1, \dots, g_n)$, $0 \leq g_i \leq 1$, ein Gewichtsvektor. a_i bzw. b_i bezeichne die Ausprägung des i -ten Attributs. Dann definieren wir das globale Ähnlichkeitsmaß durch

$$H_{g,sim}((a_1, \dots, a_n), (b_1, \dots, b_n)) = \sum_{i=1, \dots, n} g_i \cdot sim_i(a_i, b_i)$$

- Eine Verallgemeinerung könnte darin bestehen, eine Art Zugehörigkeitsfunktion für eine „fuzzy set“ zu definieren (s. Abb. links) oder eine Gewichtsfunktion zu definieren, die von Kontextparametern abhängt.



Ähnlichkeitsmaße – Zusammenfassung

- Zweck/Ziel der Ähnlichkeit:
 - ◆ Finde Fälle, die das aktuelle Problem möglichst gut darstellen und die leicht an das aktuelle Problem angepaßt werden können
 - ◆ Finde Fälle, die (fast) dieselbe Lösung wie das aktuelle Problem besitzen.
- Grundannahme: Ähnliche Fälle besitzen ähnliche Lösungen.
- Grad der Ähnlichkeit ist der Nutzwert bzw. die Wiederverwendbarkeit der Problemlösung.
- Ähnlichkeit ist eine à-priori-Abschätzung dieser Nützlichkeit bzw. Wiederverwendbarkeit
- Ziel der Ähnlichkeitsmodellierung:
 - ◆ Nähe zur wirklichen Wiederverwendbarkeit
 - ◆ leichte Berechenbarkeit

Ähnlichkeitsmaße – Zusammenfassung

- Es gibt verschiedene Ansätze der Ähnlichkeitsmodellierung, die aber von der Repräsentation der Fälle abhängen.
- Ähnlichkeitsmaß:
 - ◆ Funktion, um 2 Fälle zu vergleichen: $sim: CASE \times CASE \rightarrow [0,1]$
 - ◆ Lokales Ähnlichkeitsmaß: Ähnlichkeit auf Merkmalsebene
 - ◆ Globales Ähnlichkeitsmaß: Ähnlichkeit auf Fall- bzw. Objektebene
 - kombiniert lokale Ähnlichkeitsmaße
 - berücksichtigt die unterschiedliche Bedeutung der Attribute durch entsprechenden Gewichtungen
- Graph-Isomorphismen bei einer Repräsentation durch Graphen
- Logische Inferenzen

Die Retrieval-Phase

- Ein effizientes Retrieval ist unerlässlich, insbesondere für große Fallbasen.
- Die Retrievalstrategie hängt ab von
 - ◆ der Repräsentation der Fälle
 - ◆ der Größe der Fallbasis
- Organisation der Fallbasis:
 - ◆ lineare Listen (nur bei kleineren Fallbasen)
 - ◆ Indexstrukturen (für große bzw. größere Fallbasen)
 - KD-Bäume (Indexstrukturen für große Wissensbasen)
 - Retrievalnetze (Indexstrukturen für textuelle Fallbasen)
 - Diskriminationsnetze: bei logische Fallrepräsentation
 -
- Speicherung von Fällen
 - ◆ Datenbanken: für große Fallbasen oder bei verteilter (shared) Nutzung
 - ◆ Hauptspeicher: für kleine Fallbasen, einfache Nutzung

Die Reuse-Phase

- Adaptation der Lösung – Optionen:
- A) Keine Modifikation der Lösung : Just copy!
- B) Manuelle Adaptation durch den Benutzer
- C) Automatische Adaptation
 - ◆ Transformationsanalogie: Transformation der Lösung
 - Einsatz von Regeln bzw. Operatoren, um die Lösung geeignet anzupassen
 - setzt die Verfügbarkeit von Wissen über die Auswirkungen der vorliegenden Unterschiede der Problemstellungen voraus
 - ◆ Ableitungsanalogie: Replay of the Problem Solving Trace
 - Generative Problemlösung
 - setzt die Verfügbarkeit von Wissen über die Prinzipien der Problemlösung voraus
 - ◆ Kompositionsadaptation: Kombination verschiedener Fälle zu einer Lösung

Die Revise-Phase

- Die Revise-Phase hat bisher nur geringe Beachtung in der CBR-Theorie erfahren.
 - ◆ Oft gibt es keine explizite Revise-Phase
 - ◆ Verifikation der Lösung durch Simulation
 - ◆ Verifikation / Evaluierung der Lösung in der Realität
- Kriterien für die Revision
 - ◆ Korrektheit der Lösung
 - ◆ Qualität der Lösung
 - ◆ Andere, z.B. Präferenzen der Benutzer

Die Retain-Phase

- Lernen
- Was kann gelernt werden?
 - ◆ Neue Erfahrungen (in Form neuer Fälle)
 - ◆ Verbesserung der Ähnlichkeitsmessung, Gewichtung der Merkmale
 - ◆ Organisation bzw. Indexierung der Fallbasis
 - ◆ Erwerb von Wissen über die Lösungsadaptation
 - ◆ Vergessen von Fällen (Fälle werden nicht länger benötigt bzw. sind veraltet.)
- Methoden / Verfahren
 - ◆ Organisation der Fälle in einer Fallbasis
 - ◆ Löschen von Fällen aus der Fallbasis
 - ◆ Erklärungsbasiertes Lernen
 - ◆ Induktion, z.B. Entscheidungsbäume
 - ◆ „neuronales“ Lernen (d.h. Anwendung neuronaler Lernstrategien)

Wo speichert ein CBR-System sein Wissen?

- CBR-Systeme unterscheiden zw. 4 Wissens-Containern
 - ◆ Vokabular (Merkmale)
 - ◆ Fallbasis
 - ◆ Ähnlichkeitsmessung
 - ◆ Lösungsadaptation
- Vorteile von CBR-Systemen
 - ◆ Hohe Flexibilität
 - Das Wissen kann (anwendungsabhängig) zwischen den 4 Containern verteilt werden.
 - Im Prinzip könnte auch jeder Container das gesamte Wissen beinhalten.
 - ◆ Der Fokus liegt auf dem Wissen in der Fallbasis.
 - ◆ Das Wissen in der Fallbasis kann leicht gepflegt und angepasst werden.

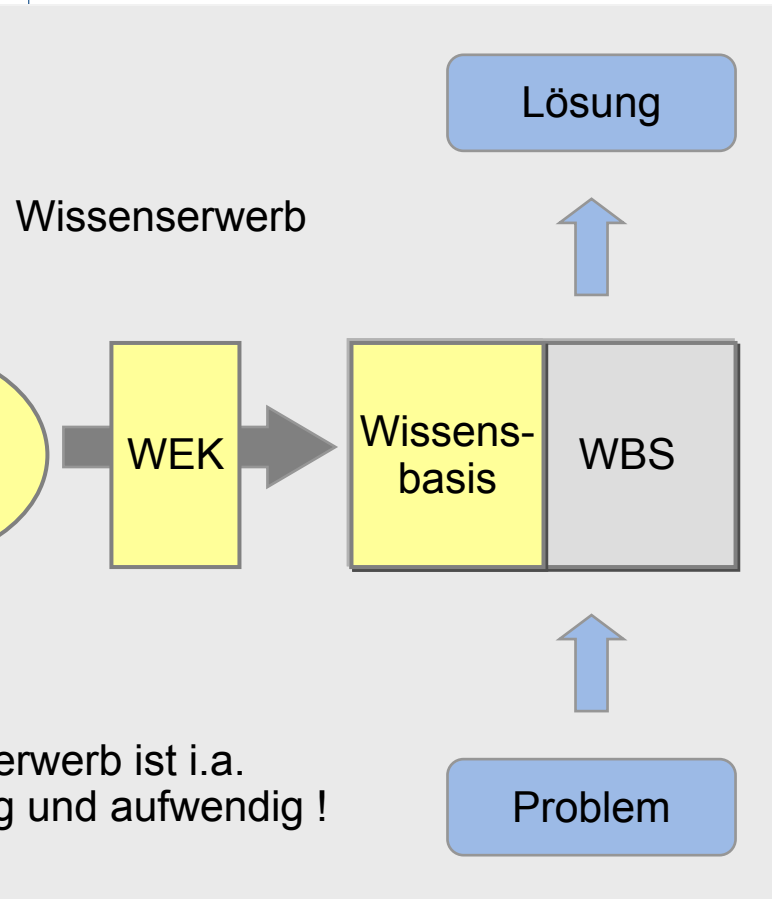
Vorteile von CBR-Systemen gegenüber anderen Verfahren

- Verringerung des Aufwandes zum Wissenserwerb
- Verringerung des Pflegeaufwandes
- Verbesserung der Problemlösung durch Wiederverwendung
- Nutzung vorhandenen Wissens bzw. vorhandener Daten in Datenbanken
- Verbesserung über die Zeit und mit Bezug auf die Änderung der Umgebung
- Hohe Akzeptanz bei Endbenutzern

Vereinfachung beim Wissenserwerb

- Typischer Wissenserwerb bei WBR

- Wissenserwerb bei CBR-Systemen



- ◆ erfordert geringeres allgemeines Wissen
- ◆ Der größte Teil des Wissens befindet sich in der Fallbasis
- ◆ Fallwissen kann leichter erworben werden oder ist oft schon vorhanden.

Aufwand für den Wissenserwerb

- Regelbasierte Systeme sind schwierig in der Pflege
 - ◆ Es gibt starke Abhängigkeiten zwischen Regeln
 - ◆ Die Regeln einer WB sind für Nicht-KI-ler oft nicht leicht zu verstehen.
 - ◆ Die Auswirkungen von Änderungen der WB sind schwer abzuschätzen.
 - ◆ Die Wartung durch Domänenexperten ist nicht immer möglich.
- Fallbasen können leichter gepflegt werden.
 - ◆ Fälle sind i.a. voneinander unabhängig.
 - ◆ Domänenexperten und Novizen verstehen neue Fälle i.a. sehr leicht.
 - ◆ Die Pflege von Fallbasen besteht i.d.R., darin, neue Fälle hinzuzufügen und alte Fälle zu entfernen.
 - ◆ Änderungen im Vokabular erfordern kaum Aufwand.

Fallbasierte Diagnose als Nächster-Nachbar-Klassifikator

Fallbasierte Diagnose als Nearest-Neighbour Klassifikation (1)

- Fallbasierte Diagnose kann man auch als einen Spezialfall der aus der Mustererkennung bekannten Nearest-Neighbour-Klassifikation auffassen.
- Zunächst einige Definitionen:

Def. 1 (Symptom): Ein Symptom ist eine meßbare Größe im technischen Sinne. Jedes Symptom $S_i = (N_i, W_i, s_i)$ besteht aus einem Namen N_i , einem Wertebereich W_i sowie einer Symptomvariablen s_i . Die Symptomvariable speichert dabei den jeweils konkret gemessenen Symptomwert $v \in W_i$. Wurde für s_i noch kein Wert erhoben, so gilt $s_i = \text{„unbekannt“}$.

Def. 2 (Situation): Als Situation (Symptomatik) Sit bezeichnen wir die zu einem Zeitpunkt t bekannten Symptomwerte (Meßwerte), d.h. $Sit \subset \{S_1, S_2, S_3, \dots, S_n\}$ mit $s_i \neq \text{„unbekannt“}$.

Def. 3 (Fall): Ein Diagnosefall (Klassifikationsfall) repräsentiert die konkrete Erfahrung eines Diagnoseexperten. Ein Fall C_k ist definiert als ein Tupel (Sit_k, D_j) , wobei Sit_k die Beschreibung der Symptomatik und D_j die in dieser Situation gestellte Diagnose ist.

Fallbasierte Diagnose als Nearest-Neighbour Klassifikation (2)

- Bezeichne C_k .Sit die einem Fall zugrundeliegende Situation und $C_k.D_j$ die in dieser Situation gestellte Diagnose. Die Diagnose $C_k.D_j$ beschreibt dabei die Klasse, der ein Fall C_k zugeordnet ist. Die Menge $FB = \{C_1, C_2, C_3, \dots, C_m\}$ heißt Fallbasis.
- Der folgende Algorithmus stellt den Kern einer fallbasierten Diagnose dar. Die Zuordnung einer Diagnose D_i zu einer Symptomatik Sit erfolgt hier, indem die Diagnose des ähnlichsten Falles C_k bzgl. eines Ähnlichkeitsmaßes sim für die aktuelle Situation übernommen wird.
- Algorithmus:
Eingabe: - Fallbasis FB
 - Ähnlichkeitsmaß sim
 - Aktuelle Situation Sit

Ausgabe: Diagnosefall C_k
 Suche in FB den Diagnosefall C_k mit:
 $sim(C_k.Sit, Sit)$ ist maximal, d.h. $\forall C_i \in FB$ gilt:
 $sim(C_i.Sit, Sit) \leq sim(C_k.Sit, Sit)$

 Gebe den Fall C_k und damit die in C_k vermerkte
 Diagnose $C_k.D$ als Lösung aus.

Fallbasierte Diagnose als Nearest-Neighbour Klassifikation (3)

- In der fallbasierten Diagnose lassen sich grundsätzlich die folgenden Ansätze unterscheiden:
- Def. 4 (Fallbasierte Diagnose):
Die Übernahme einer Diagnose D_j für eine geg. Symptomatik Sit auf der Basis eines gefundenen Fallbeispiels $C_k = (Sit_k, D_j)$ bezeichnen wir als fallvergleichende Diagnostik.
Wird die Diagnose D_j eines gefundenen Fallbeispiels für eine gegebene Symptomatik Sit modifiziert, so sprechen wir von einer fallmodifizierenden Diagnostik.
- Es handelt sich hier um einen sehr einfachen Ansatz zur fallbasierten Diagnostik.
- Stichworte für weitergehende Themen sind etwa:
 - ◆ Änderung der Fallbasis,
 - ◆ Änderung des Ähnlichkeitsmaßes,
 - ◆ Vermeidung von Fehldiagnosen,
 - ◆ wissenschaftliche Unterstützung des Fallvergleichs etc.