

# Seminararbeit im Fachbereich Wissensrepräsentation

Truth Maintenance Systeme

Autor: Elke Klippstein

1. Grundlagen
2. Truth Maintenance System (TMS)
3. Arten
  - (a) Justification-based TMS (JTMS)
  - (b) Assumption-based TMS (ATMS)
4. Vergleich verschiedener TMS
5. Ausblick

## Monotones vs. Nichtmonotones Schließen

### Monotones Schließen

- $\mathcal{F} \models \mathcal{G}$  gdw.  $\text{Mod}(\mathcal{F}) \subseteq \text{Mod}(\mathcal{G})$
- $\text{Cn} : 2^{\text{Form}} \rightarrow 2^{\text{Form}}$
- $\text{Cn}(\mathcal{F}) = \{ G \in \text{Form} \mid \mathcal{F} \models G \}$
- aus  $\mathcal{F} \subseteq \mathcal{H}$  folgt  $\text{Cn}(\mathcal{F}) \subseteq \text{Cn}(\mathcal{H})$

→ Läßt nur eine Erweiterung, keine Revidierung zu

## Monotones vs. Nichtmonotones Schließen

### Nichtmonotones Schließen

- $C : 2^{Form} \rightarrow 2^{Form}$
- $\mathcal{F} \mid \sim \mathcal{G}$  gdw.  $\mathcal{G} \subseteq C(\mathcal{F})$

→ Schlußfolgerungen zurückzunehmen und alternative Schlußfolgerungen abzuleiten

1. Grundlagen
2. Truth Maintenance System (TMS)
3. Arten
  - (a) Justification-based TMS (JTMS)
  - (b) Assumption-based TMS (ATMS)
4. Vergleich verschiedener TMS
5. Ausblick

## Aufgaben eines TMS

- Verwaltung Abhängigkeiten zwischen Aussagen oder Daten
- Erhaltung der Konsistenz bei Überführung des Systems von einen Zustand in einen anderen, bei Vorliegen neuer Informationen
- Veranlassung der Rücknahme von Aussagen, die in Konflikt mit der neuen Information steht

## Beispiel

*Schalter*     $s$  = „Schalter ist geschlossen“

$\neg s$  = „Schalter ist offen“

*Kabel*         $k$  = „Kabel ist in Ordnung“

$\neg k$  = „Kabel ist defekt“

*Lampe*         $l$  = „Lampe ist an“

$\neg l$  = „Lampe ist aus“

Regeln  $s \rightarrow l$

$s \wedge \neg k \rightarrow \neg l$

Faktenmenge  $\mathcal{F}_1 = \{s\}$ , Faktenmenge bei Kabelbruch  $\mathcal{F}_2 = \{s, \neg k\}$

widersprüchliche Faktenmenge  $\{s, \neg k, l, \neg l\}$

1. Grundlagen
2. Truth Maintenance System (TMS)
3. Arten
  - (a) Justification-based TMS (JTMS)
  - (b) Assumption-based TMS (ATMS)
4. Vergleich verschiedener TMS
5. Ausblick



## Arten TMS

- JTMS (Justification-based TMS)
  - hier darf ohne eine entsprechende Begründung nichts geglaubt oder akzeptiert werden
  - stellt konsistente Modelle zur Verfügung
- ATMS (Assumption-based TMS)
  - Berechnung und Verwaltung Mengen von Annahmen, unter denen eine Aussage ableitbar ist
  - stellt konsistente Kontexte zur Verfügung

## JTMS - Grundbegriffe

- $\mathcal{T} = (N, \mathcal{J})$ 
  - $N$  Menge endlicher Knoten,  $\mathcal{J}$  Menge von Begründungen
- $J = \langle I \mid O \rightarrow n \rangle, J \in \mathcal{J}$ 
  - $n$  ist die Konsequenz der Begründung
- $\langle \emptyset \mid \emptyset \rightarrow n \rangle$ ,  $n$  ist eine Prämisse
- $\mathcal{J}(n)$  Menge der Begründungen eines Knoten  $n \in N$
- monotone Begründung  $\langle I \mid O \rightarrow n \rangle$  mit  $O = \emptyset$
- nichtmonotone Begründung für  $O \neq \emptyset$

## Beispiel eines TMN

Sei  $\mathcal{T} = (N, \mathcal{J})$  ein TMN

Knotenmenge  $N = \{ A, B, C, D, E, F \}$

Menge der Begründungen  $\mathcal{J}$

$$J_1 = \langle C \mid \emptyset \rightarrow A \rangle$$

$$J_2 = \langle \emptyset \mid A \rightarrow B \rangle$$

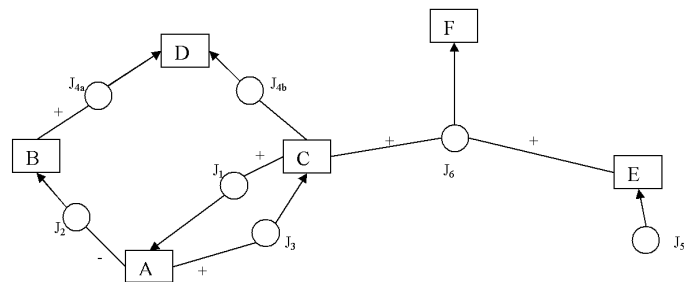
$$J_3 = \langle A \mid \emptyset \rightarrow C \rangle$$

$$J_{4a} = \langle B \mid \emptyset \rightarrow D \rangle$$

$$J_{4b} = \langle C \mid \emptyset \rightarrow D \rangle$$

$$J_5 = \langle \emptyset \mid \emptyset \rightarrow E \rangle$$

$$J_6 = \langle C, E \mid \emptyset \rightarrow F \rangle$$



## Grundlagen

- Ziel eines JTMS ist es zu einem gegebenen Netzwerk, die Etablierung eine (Wissens-)Status bzgl. der Knoten  $N$
- $N$  soll mit den gegebenen Prämissen und Begründungen konsistent sein

## Definitionen

- Status der alle akzeptierten Aussagen enthält, bezeichnet man Modell
  - Menge von Knoten  $M \subseteq N$
  - Knoten  $M$  werden als in-Knoten bezeichnet, die andern Knoten in  $N$  als out-Knoten
- Gültigkeit von Begründungen in einem Modell
  - $\langle I \mid O \rightarrow n \rangle$  ist gültig in  $M$ , wenn  $I \subseteq M$  und  $O \cap M = \emptyset$
  - Prämisse ist in jedem Modell gültig

## Definitionen

- Fundiertheit
  - bzgl.  $\mathcal{T}$ , wenn es eine vollständige Ordnung  $n_1 < \dots < n_k$  der Elemente in  $M$  gibt, es gilt  $n_j \in M$
  - es existiert in  $M$  eine gültige Begründung  $\langle I \mid O \rightarrow n_j \rangle \in \mathcal{J}$ , dass  $I \subseteq \{n_1, \dots, n_{j-1}\}$
  - schließt zirkuläre Begründungen aus
- Abgeschlossenheit einer Knotenmenge  $M$  bzgl.  $\mathcal{T}$ 
  - wenn jede Konsequenz einer in  $M$  gültigen Begründung darin enthalten ist

## Definitionen

- Abgeschlossenheit einer Knotenmenge  $M$  bzgl.  $\mathcal{T}$ 
  - es gilt für jede in  $M$  gültige Begründung  $\langle I \mid O \rightarrow n \rangle \in \mathcal{J}$ ,  
 $n \in M$
  - alle Prämissen sind Elemente eines abgeschlossenen Modells
- zulässiges Modell bzgl.  $\mathcal{T}$ 
  - TM-Netzwerk  $\mathcal{T} = (N, \mathcal{J})$  und  $M \subseteq N$
  - $M$  zulässiges Modell, wenn  $M$  fundiert und abgeschlossen bzgl.  $\mathcal{T}$  ist

## JTMS-Algorithmus

Annahme:

- M ein zulässiges Modell eines TMN
- für jedes  $n_j \in M$  wird eine stützende Begründung ausgewählt,  $SJ(n_j)$  bezeichnet
- stützende Knoten  $\text{Supp}(n)$ , eines  $n \in N$ , begründen den Status n im Modell M
- Unterscheidung der Knoten in den Status in oder out, ist er in so ist  $\text{Supp}(n) = I \cup O$ , bei out enthält  $\text{Supp}(n)$  je ein Knoten von jeder Begründung  $J \in \mathcal{J}(n)$



## JTMS-Algorithmus

Annahme:

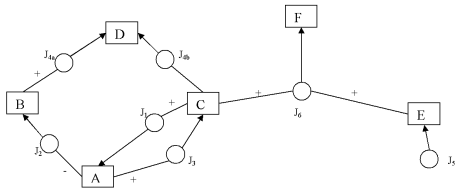
- stützende Knoten eines in-Knoten nennt man Antezedenzen von  $n$ ,  $\text{Ant}(n)$
- Knoten die  $n$  in einer ihrer Begründungen erwähnen, nennt man Konsequenzen von  $n$ ,  $\text{Cons}(n)$
- betroffene Konsequenzen von  $n$ , sind diejenigen Knoten, die bei einem Statuswechsel von  $n$  ebenfalls mit einem Statuswechsel reagieren können,  $\text{ACons}(n)$

## JTMS-Algorithmus

Annahme:

- Vorfahren eines Knoten  $n$  ist der transitive Abschluß der stützenden Knoten von  $n$ ,  $\text{Supp}^*(n)$
- Fundamente von  $n$  sind der transitive Abschluß der Antezedenzen von  $n$ ,  $\text{Ant}^*(n)$
- Auswirkungen eines Knoten  $n$  bezeichnet den transitiven Abschluß der betroffenen Konsequenzen von  $n$ ,  $\text{ACons}^*(n)$

## Beispiel



D hat A, B, C als Vorfahren  
und A, B als Fundamente

n	Status	$\mathcal{J}$	SJ(n)	Supp(n)	Supp*(n)
A	out	$J_1$	undef.	C	C, A
B	in	$J_2$	$J_2$	A	A, C
C	out	$J_3$	undef.	A	A, C
D	in	$J_{4a}, J_{4b}$	$J_{4a}$	B	B, A, C
E	in	$J_5$	$J_5$	—	—
F	out	$J_6$	undef.	C	C, A

Beispiel

n	Ant(n)	Ant*(n)	Cons(n)	ACons(n)	ACons*(n)
A	undef.	undef.	B, C	B, C	B, C, D, A, F
B	A	A	D	D	D
C	undef.	undef.	A, D, F	A, F	A, F, B, C, D
D	B	B, A	–	–	–
E	–	–	F	–	–
F	undef.	undef.	–	–	–

## weitere Festlegungen

- Fälle von Begründungen
  1. fundiert gültig - jeder Knoten der in-Liste ist in und jeder Knoten der out-Liste ist out
  2. fundiert ungültig - ein Knoten der in-Liste ist out oder ein Knoten der out-Liste ist in
  3. nicht fundiert gültig - jeder Knoten der in-Liste in ist und kein Knoten der out-Liste in ist
- trifft keine der o.g. 3 Fälle zu, dann ist die Begründung nicht fundiert ungültig

## Algorithmus JTMS-Verfahren

- Verfahren, welches nach Hinzufügen einer Begründung ein neues zulässiges Modell sucht
  1. hinzufügen von  $J_0$  und aktualisieren von  $\text{Cons}(n)$
  2. überprüfen von  $\text{ACons}(n_0)$
  3. alle Labels der Knoten in  $L$  werden unknown
  4. bestimmen des neuen Status von den Knoten in, out oder unknown
  5. untersuchen der Knoten die den Status unknown haben
  6. wenn Widerspruchsknoten vorhanden, nutze DDB
  7. liste alle Änderungen auf

## Algorithmus JTMS-Verfahren

- hinzunehmen der Menge von Widerspruchsknoten  $N_{\perp}$ ,  $\mathcal{T} = (N, N_{\perp}, \mathcal{J})$
- $M \subseteq N$  ist zulässig, wenn  $M$  zulässig und  $M \cap N_{\perp} = \emptyset$
- Modell ist nur dann zulässig, wenn kein Widerspruchsknoten in ist

## Beispiel

Modell  $M = \{ E, B, D \}$

neue Begründung, Prämisse  $A : J_0 = \langle \emptyset \mid \emptyset \rightarrow A \rangle$  und

$n_0 = A, I_0 = O_0 = \emptyset$

$$1. \mathcal{J} := \mathcal{J} + J_0$$

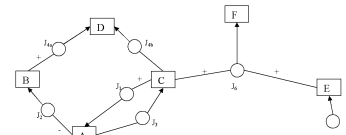
$$\mathcal{J}(A) := \{ J_1, J_0 \}$$

A out in M

$J_0$  gültig in M

$$2. \text{ACons}(A) = \{ B, C \}$$

$$3. L := \text{ACons}^*(A) + A = \{ A, B, C, D, F \}$$





## Beispiel

$L_{old} := \{ A: \text{out}, B: \text{in}, C: \text{out}, D: \text{in}, F: \text{out} \}$

Label(A) = unknown

Label(B) = unknown

Label(C) = unknown

Label(D) = unknown

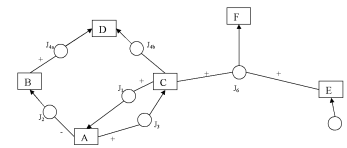
Label(F) = unknown

4.  $n^{(1)} = A$

$J_0$  ist fundiert gültig

$SJ(A) := J_0$

$\text{Supp}(A) := \emptyset$



## Beispiel

Label(A) := in

Cons(A) = { B, C }, beide den Status unknown

$n^{(11)} = B$

$\mathcal{J}(B) = \{ J_2 \}$

$J_2$  ist fundiert ungültig, da A in ist

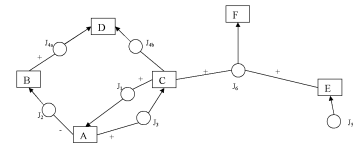
⋮

Label A, B, C, D, F sind bekannt

6. Neue Modell  $M' = \{ A, C, D, E, F \}$

Widerspruchsknoten existieren nicht

7. Knoten A, B, C, F haben ihren Status gewechselt



## Prozedur Dependency Directed Backtracking - DDB

Eingabe: TM-Netzwerk, zulässiges Modell  $M$  und Widerspruchsknoten  $n_{\perp}$  mit Status in

Ausgabe: ein zulässiges Modell bzgl. eines erweiterten TM-Netzwerks, wenn möglich

1. Bestimme die Menge  $\text{MaxAnn}(n_{\perp})$  der max. Annahmen von  $n_{\perp} \rightarrow \text{MaxAnn}(n_{\perp}) = \emptyset \rightarrow \text{HALT}$
2. Wähle eine der max. Annahmen  $n_A \in \text{MaxAnn}(n_{\perp})$ ; wähle einen (out-)Knoten  $n^*$  in der out-Liste

## Prozedur Dependency Directed Backtracking - DDB

3. Füge eine neue Begründung der Form  $\langle I_{\perp} \mid O_{\perp} \rightarrow n^* \rangle$
4. Starte JMS-Prozedur, Schritte 1-5 (wird  $n^*$  in  $\rightarrow$  ermittle neuen zulässigen Status  $M'$ )
5. Ist der Status von  $n_{\perp}$  im neuen Modell in, so wende DDB auf  $M'$  und das erweiterte Netzwerk an
6. Ist Status von  $n_{\perp}$  out  $\rightarrow$  HALT

## Beispiel - Mordverdacht

Wenn jemand Nutznießer eines Mordes ist und kein Alibi hat, so ist er verdächtig. Wenn jemand zur Tatzeit bei Freunden war und die Freunde weit entfernt vom Tatort wohnen und es angenommen werden kann, dass die Freunde nicht lügen, so hat er ein Alibi für den Mord. Es steht fest, dass Anton Nutznießer des Mordes an seiner Tante ist.

3 Begründungen:

1. Wenn jemand Nutznießer eines Mordes ist und kein Alibi hat, so ist verdächtig.

- $J_1 = \langle N \mid A \rightarrow V \rangle$

## Beispiel - Mordverdacht

2. Wenn jemand zur Tatzeit bei Freunden war und die Freunde weit entfernt vom Tatort wohnen und es angenommen werden kann, dass die Freunde nicht lügen, so hat er ein Alibi für den Mord.

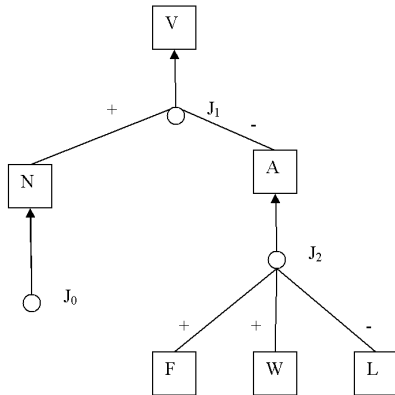
- $J_2 = \langle F, W \mid L \rightarrow A \rangle$

3. Es steht fest, dass Anton Nutznießer des Mordes an seiner Tante ist.

- $J_0 = \langle 0 \mid 0 \rightarrow N \rangle$

## Beispiel - Mordverdacht

### TMN-Entwurf



Das einzige fundierte Modell ist  $\{ N, V \}$ , da N von der Begründung  $J_0$  und V von der Begründung  $J_1$  gestützt wird.

## JTMS-Inferenzrelation

- es sind nichtmonotone Ableitungen möglich
- Trennung in Regelwissen und Faktenwissen
- Prämissen als Vorgaben regelbasiertes Wissen repräsentieren
- Annahme:  $N$  Knotenmenge,  $\mathcal{J}$  Begründungsmenge, die nur das regelhafte Hintergrundwissen enthält
- Fundiertheit wird modifiziert, um Fundiertheit bzgl. eines gegebenen Faktenwissens auszudrücken
- Repräsentierung des Faktenwissens durch eine Menge  $A$  von Knoten



## JTMS-Inferenzrelation

- Fakten werden als temporäre Prämissen des TMN betrachtet
- zulässiges Modell von  $A$  bzgl.  $\mathcal{J}$  wenn 3 Bedingungen erfüllt sind:
  1.  $A \subseteq M$
  2.  $M$  ist abgeschlossen bzgl.  $\mathcal{J}$
  3.  $M$  ist fundiert in  $A$  bzgl.  $\mathcal{J}$
- $ad_{\mathcal{J}}(A)$  ist Menge aller zulässigen Modelle einer Faktenmenge  $A$

## JTMS-Inferenzrelation

- TMS-Inferenzrelation
  - es sei  $\mathcal{T} = (N, \mathcal{J})$  ein TMN
  - $A \subseteq N$  eine Menge elementarer Aussagen
  - $b \in N$  eine elementare Aussage
  - $b$  nichtmonoton aus  $A$  folgt bzgl.  $\mathcal{J}$   $A \mid \sim_{\mathcal{J}} b$  gdw.
    - \*  $b \in \cap ad_{\mathcal{J}}(A)$  falls  $ad_{\mathcal{J}}(A) \neq \emptyset$
    - \*  $b \in A$  falls  $ad_{\mathcal{J}}(A) = \emptyset$

## Beispiel

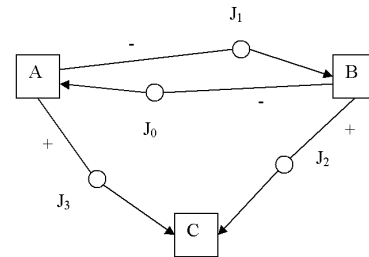
Knotenmenge  $N$ , aus  $A$ ,  $B$  und  $C$

Begründungen:

$$J_0 : \langle \emptyset \mid B \rightarrow A \rangle$$

$$J_1 : \langle \emptyset \mid A \rightarrow B \rangle$$

$$J_2 : \langle A \mid \emptyset \rightarrow C \rangle$$

$$J_3 : \langle B \mid \emptyset \rightarrow C \rangle$$


$ad_{\mathcal{J}}(\emptyset)$  hat zwei Elemente  $\{ A, C \}$  und  $\{ B, C \}$   
 der Schnitt ist  $\cap ad_{\mathcal{J}}(\emptyset) = \{ C \}$ , daraus folgt  $\emptyset \mid \sim_{\mathcal{J}} C$

## Assumption-based TMS

- Terminologie ähnlich zum JTMS
- Knoten repräsentieren Aussagen und Begründungen werden mit derer Aussagen abgeleitet
- jeder Knoten entspricht einer atomaren Aussage
- Begründung hat die Form  $n_1, n_2, \dots \rightarrow n$  entspricht einer materialen Implikation  $n_1 \wedge n_2 \wedge \dots \rightarrow n$
- keine Unterscheidung in in- und out-Knoten
- Annahmen werden durch Knoten repräsentiert  $A \rightarrow n$

## Assumption-based TMS

- Menge von Annahmen wird als Umgebung bezeichnet
- $n$  ist gültig in  $E$  gdw.  $E, \mathcal{J} \vdash n$
- eine Umgebung ist inkonsistent, wenn aus ihr auf diese Weise das Falsum  $\perp$  abgeleitet werden kann
- Kontext ist eine Menge von Knoten, die in einer Umgebung gültig ist

## Arbeitsweise eines ATMS

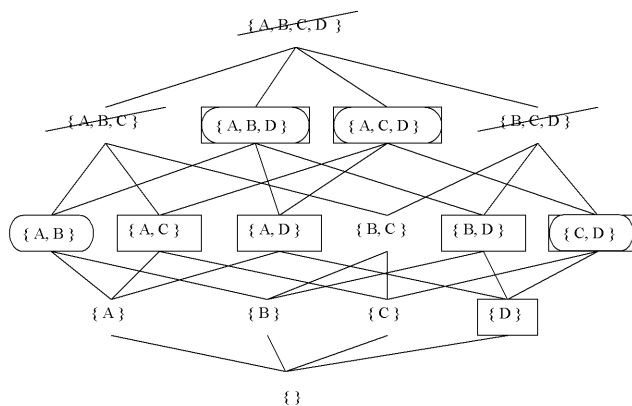
- Bestimme alle Kontexte - aus Effizienzgründen nicht wörtlich erfüllbar
- Aufdeckung von Inkonsistenzen in Kontexten
- Entscheidung, ob ein Knoten in einem bestimmten Kontext enthalten ist
- Knoten  $n$  wird in der Form  $n = \langle \textit{Aussage}, \textit{Label}, \textit{Begründungen} \rangle$  dargestellt

## Arbeitsweise eines ATMS

- Label besitzt die Eigenschaften
  - es ist konsistent, d.h. alle seine Umgebungen sind konsistent
  - es ist korrekt für den Knoten  $n$ , d.h.  $n$  ist aus jeder der Umgebungen des Labels ableitbar
  - es ist vollständig, d.h. jede konsistente Umgebung, in der  $n$  gültig ist, ist eine Obermenge der Labelmenge
  - es ist minimal, d.h. keine der Labelmengen ist in einer anderen enthalten

## Umgebungsverband

Verband aller Teilmengen der Menge  $\{ A, B, C, D \}$



Ovale sind die konsistente  
Umgebung für  $n_1$

Rechtecke konsistente  
Umgebung für  $n_2$

$n_1 = \langle a, \{ \{ A, B \}, \{ C, D \} \}, \dots \rangle$

$n_2 = \langle b, \{ \{ A, C \}, \{ D \} \}, \dots \rangle$

$n_3 = \langle c, \{ \{ C, D \}, \{ A, B, D \} \}, \{ n_1, n_2 \rightarrow n_3 \} \rangle$



## Umgebungsverband

### Bestimmung der Labelmengen

- Korrektheit und Vollständigkeit - Bildung der Vereinigung aus je einer Labelmenge eines jeden Antezedenzknoten und berücksichtigt alle möglichen Kombinationen
  - man erhält die Mengen  $\{ A, B, C \}$ ,  $\{ A, B, D \}$ ,  $\{ A, C, D \}$ ,  $\{ C, D \}$  für  $n_1$  und  $n_2$
- Minimalität und Konsistenz - Streichung inkonsistente Umgebungen
  - übrig gebliebene Mengen  $\{ A, B, D \}$  und  $\{ C, D \}$

1. Grundlagen
2. Truth Maintenance System (TMS)
3. Arten
  - (a) Justification-based TMS (JTMS)
  - (b) Assumption-based TMS (ATMS)
4. Vergleich verschiedener TMS
5. Ausblick

## Vergleich verschiedener TMS

- ATMS
  - entspricht einer Breitensuche
  - Überblick über alle möglichen Lösungen
  - Ergebnisse eines Kontextwechsels können sofort repräsentiert werden
  - entfernt alle Inkonsistenzen von vornherein
  - analysiert keine logischen Strukturen

## Vergleich verschiedener TMS

- JTMS
  - entspricht der Tiefensuche
  - Darstellung und Behandlung nicht-monotoner Zusammenhänge
  - hinzufügen von Annahmen ist weniger problematisch
  - analysiert keine logischen Strukturen
- LTMS
  - analysiert logische Strukturen

1. Grundlagen
2. Truth Maintenance System (TMS)
3. Arten
  - (a) Justification-based TMS (JTMS)
  - (b) Assumption-based TMS (ATMS)
4. Vergleich verschiedener TMS
5. Ausblick

## Ausblick

- weißt enge Beziehungen zur Default-Logik und zum Logischen Programmieren auf
- das JTMS-Verfahren kann nur angewendet werden, wenn einem TM-Netzwerk Begründungen hinzugefügt werden, nicht aber wenn sie entfernt oder modifiziert werden
- das DDB-Verfahren kann nur erfolgreich durchgeführt werden, wenn Annahmen zurückgesetzt werden können