

# Technologie zur dynamischen Analyse von Lernzielstrukturen

Matthias Weigt  
email: matthias.w.weigt@gmail.com

3. März 2023

## **Vorwort**

Eine der größten Herausforderungen beim Lernen ist, meiner Auffassung nach, das Auswählen von Inhalten mit der höchsten Relevanz für das Erreichen eines bestimmten Ziels.

Solche Inhalte sind oftmals in Eigenreflexion schwierig auszumachen und können auch durch eine systematische Analyse nicht leicht sichtbar gemacht werden. Gerade das Einordnen von Kenntnissen in eine Systematik, ist umso schwieriger für einen Laien, desto weniger dieser die größere Systematik erfasst hat.

Im Folgenden lege ich ein Technologiekonzept dar, welches bei diesem Problem zu helfen vermag. Das Konzept umfasst unter anderem:

- (i) die mathematische Abbildung von Zusammenhängen zwischen Lernzielen
- (ii) verschiedene Konstruktionsmöglichkeiten für die entstehenden Strukturen
- (iii) die Analyse der Zusammenhänge der Strukturen
- (iv) verschiedene Testverfahren zur Abbildung von Kenntnisständen von Personen auf diese Strukturen

## **WICHTIG:**

Diese Ausarbeitung hat (noch) nicht den Anspruch einem wissenschaftlichen Standard zu entsprechen. Ich gebe im Folgenden keine Quellen an, alle angegebenen Verfahren sind von mir selbst oder in Kooperation mit meinen Partner:innen entwickelt worden.

## **Korrektur und Nachtrag:**

Dieses Skript wurde korrigiert und nachgetragen von Iris Weigt und Mark Goldmann.

# Inhaltsverzeichnis

<b>1</b>	<b>Kerntechnologie - Lernzielstrukturen</b>	<b>1</b>
1.1	Lernziel . . . . .	1
1.2	Lernzielstruktur . . . . .	1
1.2.1	Logische Betrachtung der Lernzielabhängigkeit . . . . .	1
1.2.2	Beispiel . . . . .	1
1.2.3	Teil-Lernzielstruktur . . . . .	2
1.3	Tiefe eines Lernziels . . . . .	2
1.4	Dependenz-Koeffizient . . . . .	2
1.5	Lernzielstruktur-Konstruktion . . . . .	2
1.6	Testen von Lernzielstrukturen . . . . .	3
1.6.1	Testmechanismen von Lernzielen . . . . .	3
1.6.2	Testverfahren . . . . .	3
1.6.3	Testverfahren Typ $\alpha$ . . . . .	3
1.6.4	Testverfahren Typ $\beta$ . . . . .	3
1.6.5	Testverfahren Typ $\gamma$ . . . . .	4
1.6.6	Testverfahren Typ $\delta$ . . . . .	4
1.7	Lernzielstruktur-Rekonstruktion . . . . .	5
1.7.1	Lernziel-Test-Log (LTL) . . . . .	5
1.7.2	Abhängigkeitsmaß . . . . .	6
1.7.3	Der Abhängigkeitsschwellwert . . . . .	6
1.7.4	Rauschen . . . . .	6
1.7.5	Streichen von redundanten Elementen aus $R \in S$ . . . . .	6
1.8	Hybride Lernzielstruktur-Konstruktion . . . . .	7
1.8.1	Hybride Konstruktion . . . . .	7
1.9	Äquivalenz von Test- und Übungsverfahren . . . . .	7
1.9.1	Testverfahren - erste Anwendung . . . . .	7
1.9.2	dynamischer Testmechanismus . . . . .	7
1.9.3	Testverfahren - weitere Anwendung . . . . .	8
1.9.4	Langzeit dynamische Testmechanismen . . . . .	8

# 1 Kerntechnologie - Lernzielstrukturen

## 1.1 Lernziel

Ein Lernziel ist eine bestimmte Fähigkeit, die eine Person zu einem gewissen Grad beherrschen kann. Wir nehmen an, dass sich Lernziele vereinfacht als aussagenlogische Aussagen betrachten lassen. „wahr“ bedeutet in diesem Fall, dass das Lernziel beherrscht wird und „falsch“ das es nicht beherrscht wird. Der Einfachheit wegen schreiben wir im Folgenden für alle Lernziele  $a$ :

- (i)  $a \Leftrightarrow$  "Lernziel  $a$  wird beherrscht."
- (ii)  $\neg a \Leftrightarrow$  "Lernziel  $a$  wird nicht beherrscht."

## 1.2 Lernzielstruktur

Eine Lernzielstruktur  $S$  ist ein 2-Tupel aus einer Menge  $L$  an Lernzielen und einer Relation  $R$  bestehend aus einer Menge geordneter Lernzielpaare mit  $S = (L, R)$ .

Ein Graph zu  $S$  soll gerichtet und nicht zyklisch sein. Es müssen somit folgende Eigenschaften für  $S$  erfüllt sein:

- (i) Es existiert keine Folge  $F$  an Elementen aus  $L$  mit  $(x_1, [\dots, x_n])$  mit  $x_1 = x_n$ .  
Somit folgen:
- (ii)  $\forall a, b \in L : (a, b) \in R \Rightarrow (b, a) \notin R$
- (iii)  $\forall a \in L : (a, a) \notin R$

Die Knoten des Graphen repräsentieren die Lernziele und eine gerichtete Kante die Beziehung zwischen zwei Lernzielen. Ein Knoten kann auf  $n$  weitere Knoten verweisen und jeder Knoten kann  $n$  Eingangskanten haben. Eine Beziehung zeigt über  $n$  Kanten, welche Lernziele benötigt werden, um ein anderes Lernziel zu erlernen. Der Begriff „Beziehung“ wird im Weiteren durch den der „Abhängigkeit“ ersetzt.

### 1.2.1 Logische Betrachtung der Lernzielabhängigkeit

Um genauer zu verstehen, wie unserer Lernzielabhängigkeit definiert ist, bedienen wir uns der aussagenlogischen Schreibweise:

Sei das Tupel  $(a, b) \in R$  so gilt:  $a \Rightarrow b$  bzw.  $\neg b \Rightarrow \neg a$

- (i) Wenn Lernziel  $a$  beherrscht wird, muss auch  $b$  beherrscht werden.
- (ii) Wenn Lernziel  $b$  nicht beherrscht wird, darf auch  $a$  nicht beherrscht werden.

Aussagenlogisch gilt: (i) $\Leftrightarrow$ (ii)

### 1.2.2 Beispiel

Sei  $L := \{a, b, c, d, e\}$  eine Menge an Lernzielen,  $R := \{(a, b), (a, c), (b, c), (b, d), (b, e)\}$  eine Relation, so ist die Lernzielstruktur  $S = (L, R)$  valide.

Aus  $R$  lässt sich ablesen, dass die Lernziele  $b, c$  für das Lernziel  $a$  beherrscht werden müssen. Für  $b$  benötigt man die Lernziele  $c, d$  und  $e$ . Das Lernziel  $c$  verweist auf keine weiteren Lernziele, da  $c$  kein Teil des Vorbereichs von  $R$  ist. Die Gesamtmenge  $G$  aller Lernziele, die beherrscht werden müssen, um  $a$  erlernen zu können ist somit  $G := \{b, c, d, e\}$ . Das bedeutet nicht, dass  $a$  erlernt wurde, wenn  $G$  beherrscht wird.

$G$  ist somit eine **notwendige**, aber nicht eine **hinreichende** Bedingung für das Beherrschen von  $a$ .

### 1.2.3 Teil-Lernzielstruktur

Eine Teil-Lernzielstruktur  $T$  ist eine Lernzielstruktur mit einer weiteren Eigenschaft:

- (i)  $\exists a \in L : \forall b \in L : b \neq a \Rightarrow \exists \text{ Folge } F \text{ aus Elementen aus } L \text{ mit } F := (a, [...], b)$ .
- (ii) Das  $a$  aus der ersten Bedingung nennen wir **Stamm**.

Aus (i) und (ii) ist ersichtlich, dass es in  $L$  immer ein Lernziel gibt von welchem aus über Kanten jedes andere Lernziel in  $T$  erreicht werden kann. Eine Teil-Lernzielstruktur kann zu jedem Lernziel  $a$  einer vorhandenen Lernzielstruktur gebildet werden, indem alle abhängigen Lernziele von  $a$  und alle zugehörigen Tupel aus  $R$  übernommen werden.

Sei  $S = (L, R)$  eine Lernzielstruktur,  $S' = (L', R')$  eine Teil-Lernzielstruktur von  $S$  somit gilt:

- (i)  $L' \subseteq L$
- (ii)  $R' \subseteq R$

### 1.3 Tiefe eines Lernziels

Jedes Lernziel  $a \in L$  innerhalb einer Teil-Lernzielstruktur hat eine Tiefe  $\text{tiefe}(a)$ . Die **Tiefe** gibt an wie viele Lernziele maximal durchquert werden müssen, um zu einem Lernziel ohne Ausgangskanten zu gelangen. Für die äußersten Lernziele (ohne Ausgangskanten)  $z_1, z_2, \dots, z_n \in L$  einer Teil-Lernzielstruktur gilt  $\text{tiefe}(z_1) = \text{tiefe}(z_2) = \text{tiefe}(z_n) = 0$ , sie werden **Blätter** der Teil-Lernzielstruktur genannt. Die Lernziele, die keine Blätter und nicht der Stamm einer Lernzielstruktur sind, heißen **Äste**. Wir nennen die gesamte Lernzielstruktur auch **Baum**.

### 1.4 Dependenz-Koeffizient

Seien  $T$  eine Teil-Lernzielstruktur,  $L_0 := \{l | l \in L : \text{tiefe}(l) = 0\}$  dessen Menge an Blättern und  $f : L_0 \rightarrow \mathbb{N}$  eine Abbildung, die jedem Element aus  $L_0$  die Anzahl an Knoten in  $T$  zuordnet, die maximal nötig sind, um zum Stamm von  $T$  zu kommen.

Der **Dependenz-Koeffizient**  $dc(T)$  gibt an, wie stark die Lernziele in  $T$  aufeinander aufbauen.

$$dc(T) := \frac{\sum_{l \in L_0} f(l)}{|L_0|}$$

$dc(T)$  ist somit der durchschnittliche, maximale Weg eines Blattes zum Stamm.

### 1.5 Lernzielstruktur-Konstruktion

Es ist möglich eine Lernzielstruktur durch eigene Kenntnisse von Grund auf zu konstruieren. Dafür sind folgende Schritte notwendig:

- (i) Identifikation der Lernziele  $L$
- (ii) Identifikation der Relation  $R$

Um in einem Thema eine solche Lernzielstruktur zu konstruieren, ist **sehr umfangreiches** Domainwissen erforderlich. Es ist somit häufig schwierig eine umfassende, korrekte Lernzielstruktur zu konstruieren, die zusätzlich einen hohen Dependenz-Koeffizienten aufweist.

## 1.6 Testen von Lernzielstrukturen

Sei  $T = (L, R)$  eine Teil-Lernzielstruktur,  $P$  eine Person.

Wir wollen für  $P$  herausfinden, wie alle Lernziele innerhalb von  $L \in T$  beherrscht werden.

### 1.6.1 Testmechanismen von Lernzielen

Sei  $L$  eine Menge an Lernzielen.

Nach 1.1 gibt es für jedes Lernziel  $l \in L$  zwei Aussagen. Das Belegen eines Lernziels mit einer Aussage ist der Mechanismus des Testens. Ein Testmechanismus **TM** ist somit eine Abbildung  $f : L \rightarrow [0, 1]$ , die für jedes Lernziel  $l$  einen Wert aus  $[0, 1]$  zurückgibt.

### 1.6.2 Testverfahren

Sei  $\mathcal{L}$  eine Menge an Teil-Lernzielstrukturen.

Ein **Testverfahren TP** ist eine Abbildung  $g : \mathcal{L} \rightarrow \mathcal{L}$ , bei der jedem Lernziel ein Beherrschungsgrad als Attribut zugewiesen wird. Diese Abbildung ist somit als ein Homomorphismus zu verstehen, welcher die Struktur in  $\mathcal{L}$  erhält.

### 1.6.3 Testverfahren Typ $\alpha$

Sei  $T = (L, R)$  eine Teil-Lernzielstruktur,  $P$  eine Person.

Bei dem  $\alpha$  Testverfahren werden mit beliebigen Testmechanismen, nacheinander alle Lernziele aus  $L$  auf Beherrschung getestet.

Stärken:

- (i) Es werden keine Lernziele aufgrund von Strukturdefiziten von  $T$  ausgelassen.

Schwächen:

- (i) Wenn  $P$  nicht viele Lernziele beherrscht, werden potenziell Lernziele getestet, bei denen  $R$  aus-sagen würde,  $P$  kann sie nicht beherrschen.

### 1.6.4 Testverfahren Typ $\beta$

Sei  $T = (L, R)$  eine Teil-Lernzielstruktur,  $P$  eine Person.

Bei dem  $\beta$  Testverfahren werden mit beliebigen Testmechanismen, nacheinander alle Lernziele aus  $L$  mit aufsteigender *tiefe*( $a \in L$ ) auf Beherrschung getestet.

Stärken:

- (i) Es werden keine Lernziele aufgrund von Strukturdefiziten von  $T$  ausgelassen.
- (ii) Es wird mit "leichteren" Lernzielen begonnen und mit "schwierigeren" fortgefahren.

Schwächen:

- (i) Wenn  $P$  nicht viele Lernziele beherrscht, werden potenziell Lernziele getestet, bei denen  $R$  aus-sagen würde,  $P$  kann sie nicht beherrschen.
- (ii) Je kleiner  $dc(T)$  ist, desto ähnlicher ist  $\beta$  zu  $\alpha$ .

### 1.6.5 Testverfahren Typ $\gamma$

Sei  $T = (L, R)$  eine Teil-Lernzielstruktur,  $P$  eine Person.

Bei dem  $\gamma$  Testverfahren wird, während des Testens auf die bis dahin entstandenen Ergebnisse zum weiteren Testen zurückgegriffen. Das Testverfahren basiert auf dem Konzept der dynamischen Programmierung.

Ablauf:

- (i) Es wird ein Index  $n = 0$  gesetzt.
- (ii) Eine Menge  $L_n := \{l | l \in L : tiefe(l) = n\}$  wird gebildet.
- (iii)  $\forall l \in L_n$  wird überprüft, ob  $l$  von einem Lernziel abhängig ist bei dem die Beherrschung gleich 0 ist.  
All diese Lernziele werden aus  $L_n$  entfernt.
- (iv) Abbruch, wenn  $L_n = \emptyset$
- (v)  $\forall l \in L_n$  wird ein beliebiger Testmechanismus angewendet.
- (vi)  $n$  wird um 1 erhöht.
- (vii) Es wird bei (ii) fortgefahren.

Stärken:

- (i) Es werden nur so viele Lernziele getestet wie nötig.
- (ii) Es wird mit "leichteren" Lernzielen begonnen und mit "schwierigeren" fortgefahren.
- (iii) Es werden nur Lernziele getestet, die  $P$  hypothetisch beherrschen könnte.

Schwächen:

- (i) Je ungenauer der verwendete Testmechanismus oder je mehr Strukturdefizite  $T$  hat, desto ungenauer ist das Endergebnis, da Lernziele weggelassen werden, die real beherrscht werden könnten.

### 1.6.6 Testverfahren Typ $\delta$

Sei  $T = (L, R)$  eine Teil-Lernzielstruktur,  $P$  eine Person.

Bei dem  $\delta$  Testverfahren wird, während des Testens auf die bis dahin entstandenen Ergebnisse zum weiteren Testen zurückgegriffen. Das Testverfahren basiert ebenfalls auf dem Konzept der dynamischen Programmierung. Zusätzlich wird eine Schrittlänge  $\lambda$  eingeführt, die das Testen durch den Baum verschnellert.  $\lambda$  sollte umso größer gewählt werden, desto größer  $dc(T)$  ist. Im Unterschied zum  $\gamma$  Testing wird allerdings ein Topdown und kein BottomUp Ansatz verwendet.

Ablauf:

- (i) Es wird eine Schrittlänge  $\lambda$  gewählt.
- (ii) Die zu testende Person  $P$  wird nach ihrem Kenntnisstand  $k$  des Themas durch Selbst- oder Fremdeinschätzung eingeschätzt.
- (iii) Es wird ein Index  $n \in [MIN(\{tiefe(l) | l \in L\}), MAX(\{tiefe(l) | l \in L\})]$  gesetzt, der Index wird umso höher gewählt, desto größer der geschätzte Kenntnisstand von  $P$  ist.

- (iv) Eine Menge  $L_n := \{l | l \in L : tiefe(l) = n\}$  wird gebildet.
- (v)  $\forall l \in L_n$  wird überprüft, ob  $l$  von einem Lernziel abhängig ist bei dem die Beherrschung gleich 0 ist.  
All diese Lernziele werden aus  $L_n$  entfernt.
- (vi) Abbruch, wenn  $L_n = \emptyset$
- (vii) Es werden alle Lernziele aus  $L_n$  getestet.
- (viii) Wenn ein Lernziel  $l \in L_n$  beherrscht wird, werden auch alle von  $l$  abhängigen Lernziele als beherrscht markiert.
- (ix) Wenn  $l$  nicht beherrscht wird, werden die weiteren abhängigen Lernziele aus  $R(l)$  nach dem gleichen Verfahren getestet.
- (x)  $n$  wird erhöht um  $MIN(\lambda, MAX(\{tiefe(l) | l \in L\}) - n + 1)$
- (xi) Es wird bei (iv) fortgefahren.

Stärken:

- (i) Das Endergebnis ist gleich genau, wie beim  $\gamma$  Testen.
- (ii) Es werden tendenziell weniger Lernziele getestet als bei  $\gamma$ .
- (iii) Es ist ggf. das effizienteste Testverfahren.

Schwächen:

- (i) Je ungenauer der verwendete Testmechanismus oder je mehr Strukturdefizite  $S$  hat, desto ungenauer ist das Endergebnis, da Lernziele weggelassen werden, die real beherrscht werden könnten.
- (ii) Je kleiner  $k$ , desto ähnlicher ist  $\delta$  zu  $\gamma$ .  
Den einzigen Unterschied macht bei einem kleinen  $k$ , die Schrittlänge  $\lambda$ .  
Bei einem sehr kleinen  $k$  kann beim  $\gamma$  Testing, kann die Anzahl der zu testenden Lernziele kleiner sein als beim  $\delta$  Testing.

## 1.7 Lerzielstruktur-Rekonstruktion

In diesem Abschnitt bedienen wir uns an der Disziplin der Geschäftsdatenanalyse.

**Idee aus dem Process-Mining** : Process-Mining wendet Data-Science an, um Prozesse zu visualisieren, zu rekonstruieren, oder um sie zu verbessern.

Hierbei werden von einem Prozess alle möglichen Log-Daten erhoben und mithilfe von Algorithmen ausgewertet.

### 1.7.1 Lerziel-Test-Log (LTL)

Ein **LTL** ist ein Log an Ausführungen von Testmechanismen. Es sollte folgende Attribute umfassen:

- (i) Id  $id$  (Primärschlüssel)
- (ii) IdLernziel  $l$

- (iii) IdPerson  $p$
- (iv) IdTestmechanismus  $tm$
- (v) IdTestverfahren  $tv$
- (vi) IdTest  $tId$  (Testdurchlauf)
- (vii) Level  $lv$  des Lernziels
- (viii) Zeitstempel  $t$

Ein **LTL** ist somit eine Menge an Tupeln der Form  $(id, l, p, tm, tv, tId, lv, t)$ .

### 1.7.2 Abhängigkeitsmaß

Wir wissen aus 1.2.1, wie die Lernzielabhängigkeit definiert wurde.

Weiterhin wollen wir eine Rechenweise einführen, womit sich diese Abhängigkeit errechnen lässt. Seien  $\mathfrak{L}$  ein *LTL*,  $L$  die Menge an Lernzielen in  $\mathfrak{L}$  und  $a, b \in L$

Das **Abhängigkeitsmaß**  $\mu(a, b)$  gibt an, ob  $a$  von  $b$  abhängt. Es ist definiert durch:

$$\mu(a, b) := \begin{cases} \frac{|\neg a \wedge b| + |a \wedge \neg b|}{|a \neq b|} & |a \neq b| > 0 \\ 0 & |a \neq b| = 0 \end{cases}$$

Anhand dieser Formel lassen sich nun sämtliche Abhängigkeiten zwischen den Lernzielen bestimmen. Je größer das LTL und je genauer die Testmechanismen, desto genauer die resultierende Abhängigkeit.

### 1.7.3 Der Abhängigkeitsschwellwert

Der **Abhängigkeitsschwellwert**  $\sigma$  gibt an, ab welchem Wert eine Abhängigkeit nach einem Testverfahren in die Relation  $R$  einer Lernzielstruktur übernommen werden soll. Die Struktur, die aus den Abhängigkeiten resultiert, wird durch das Setzen von  $\sigma$  maßgeblich bestimmt. Bei sehr großen LTLs ist ein Wert nahe an Eins zu empfehlen.

### 1.7.4 Rauschen

Rauschen im LTL entsteht durch minderwertige Testmechanismen oder zufällig auftretende Eingaben der Person, die die eigentlichen Fähigkeiten der Person nicht korrekt abbilden. Das Rauschen ist in kleinen LTLs häufig **sehr hoch**. Die Probleme, die durch Rauschen auftreten, lassen sich durch drei Faktoren minimieren:

- (i) Die Verbesserung der Testmechanismen
- (ii) Die Vergrößerung des LTLs
- (iii) Die Anpassung des Schwellwerts  $\delta$

### 1.7.5 Streichen von redundanten Elementen aus $R \in S$

Sei  $S = (L, R)$  eine Lernzielstruktur.

Das Rekonstruieren von Lernzielstrukturen anhand von  $\mu$  erzeugt redundante Tupel innerhalb der Relation  $R \in S$ .

Sei  $L \in S := \{a, b, c\}$ ,  $R := \{(a, b), (a, c), (b, c)\}$ .

Dann ist das Tupel  $(a, c)$  redundant, da die Abhängigkeit von  $c$  zu  $a$  bereits über die Tupel  $(a, b), (b, c)$  gezeigt ist.

Das Tupel  $(a, c)$  ist somit bei der Rekonstruktion von  $S$  aus  $R$  zu entfernen.



- (i) Die Relation  $R$  darf nicht transitiv sein.
- (ii) Alle "Kandidaten" für Transitivität sind aus der Relation bei der Rekonstruktion zu entfernen.

## 1.8 Hybride Lernzielstruktur-Konstruktion

Sowohl die Konstruktion von Lernzielstrukturen 1.5, als auch die Rekonstruktion von Lernzielstrukturen 1.7.4 weist einige Schwierigkeiten auf.

- (i) Für die **Konstruktion** von Lernzielstrukturen wird ein **hohes Domainwissen** und **hoher Arbeitsaufwand** benötigt.
- (ii) Für die **Rekonstruktion** von Lernzielstrukturen wird ein **großes LTL** benötigt, da sonst das **Rauschen** zu stark ist.

Um die Probleme aus (i) und (ii) zu minimieren ist es möglich eine **hybride Konstruktion** für die Lernzielstruktur zu verwenden.

### 1.8.1 Hybride Konstruktion

- (i) Wir beginnen mit der Konstruktion nach 1.5 einer Lernzielstruktur  $S$ .  
Dabei gehen wir allerdings nicht zu detailliert vor.
- (ii) Wir verwenden das  $\beta$  Testverfahren 1.6.4 um ein LTL zu erstellen.
- (iii) Wir verwenden Lernzielstruktur Rekonstruktion 1.7 anhand des LTLs.
- (iv) Wir gleichen das neue Modell mit dem alten Modell ab und verbessert es, indem wir neue Tupel in die Relation  $R \in S$  hinzufügen oder welche streichen.

## 1.9 Äquivalenz von Test- und Übungsverfahren

Es ist anzunehmen, dass einer Person  $P$ , nachdem sie ein Testverfahren durchlaufen hat und nun ihren Kenntnisstand kennt, sich einige der noch fehlenden Lernziele aneignen möchte.

Wir sind bereits bei dem  $\gamma$ -Testen (siehe 1.6.5) auf ein dynamisches Testverfahren eingegangen.

Wir können allerdings das Konzept der dynamischen Programmierung nicht nur auf die einzelnen *Tiefen* der Lernziele in einem Test anwenden, sondern auch über komplette Testdurchläufe hinweg.

### 1.9.1 Testverfahren - erste Anwendung

Bei der ersten Anwendung eines Testverfahrens auf  $P$  ist noch nichts bekannt über die Kenntnisse von  $P$ . Es müssen somit, je nach Typ des Testverfahrens viele Lernziele überprüft werden.

Die Ergebnisse, die die Testmechanismen liefern, werden in einem LTL gespeichert.

### 1.9.2 dynamischer Testmechanismus

Ein **dynamischer Testmechanismus (DTM)** ist ein Testmechanismus, der auf ein LTL zugreifen kann.

Ein DTM kann Algorithmen beinhalten, die unter bestimmten Umständen ein Lernziel als "beherrscht" einstufen, ohne es abzufragen. Das kann z.B. sinnvoll sein, wenn:

- (i) die letzte Testung nicht lange zurückliegt und das Lernziel bereits letztes Mal beherrscht wurde.
- (ii) aufbauende Lernziele bei dem letzten Testverfahren beherrscht wurden.

**Beispiel:**

Es ist nicht sinnvoll abzufragen, ob  $P$  Polynome 2. Grades ableiten kann ( $B$ ), wenn  $P$  verknüpfte Funktionen aus Polynomen und trigonometrischen Funktionen mithilfe der Produktregel ableiten kann ( $A$ ), denn  $A \Rightarrow B$ .

**1.9.3 Testverfahren - weitere Anwendung**

Bei jeder weiteren Anwendung eines Testverfahrens z.B. zur Übung, können die DTMs auf das LTL von  $P$  zurückgreifen.

Es werden somit überwiegend Lernziele abgefragt, die sehr genau auf den Kenntnisstand von  $P$  zugeschnitten sind.

Dadurch spart  $P$  Zeit und der Übungs- und Lernaspekt steigt.

**1.9.4 Langzeit dynamische Testmechanismen**

Ein **Langzeit dynamischer Testmechanismus LDTM** ist ein DTM, der die LTLs von mehreren vergangenen Testverfahren berücksichtigt.

Er kann z.B. auch die Folge an richtigen Outputs eines Lernziels  $F$  mit in seine Berechnung einbeziehen, die aussagt, ob es Abgefragt werden soll oder nicht. Das kann z.B. an eine Funktion geknüpft sein, die  $F$  als Parameter akzeptiert und die Zeit, nach der das Lernziel erneut abgefragt werden soll zurückgibt. Damit könnte sichergestellt werden, dass früher gekonnte Inhalte nicht in Vergessenheit geraten und nach immer größer werdenden Abständen erneut wiederholt werden.

Eine Kombination von (i) und (ii) im obigen Beispiel (siehe 1.9.2) auf die Weise eines LDTMs liefert dabei das effizienteste Ergebnis.