

知識の自由探訪を支援するシステム

伊藤 紘二 浅野 正一郎 堀 浩一 横尾 真

1. はじめに

生涯学習における独習を援助する計算機利用システムにおいては、つきの2点が要請される。

- (1) 広範な諸分野にまたがる知識を、学習者に既知な概念からアクセスし、段階的に習得してゆくことができるここと、
- (2) 学習者が当面する問題の解決に役立つ知識とその適用方法を、手早く学習出来ること。

本稿は、このような要請を、知識情報処理の現状ならびにシステム資源の有効利用と両立させるために考えられる1つのアプローチを提案する。

2. 提案するシステムの構想

システムは、

- (i) 文章、図、プログラムなどにより記述されるユニット化されたテキストの集まり、
 - (ii) これらのテキストの検索と提示の制御を支援する機構を持つ知識ベース、
 - (iii) 知識ベースを制御する機構、
- とからなる。

知識ベースは、

- ① 学習者との間に用語レベルでインターフェースをとるための用語リスト (Term List - TML)
- ② 記述的、論理的、因果的等、さまざまなカテゴリの知識の単位について、テキストへのポインタをもち、学習者との対話に基づいてテキストの検索の支援と提示の制御を行なうための知識記述ユニット (Knowledge Description Unit - KDU)
- ③ KDUの記述の枠組みを与える概念フレームユニット (Conceptual Frame Unit - CFU)

④ CFUの間の階層的、構造的等、諸関係を記述し、それに基づいて、簡単な推論を行うことによって KDU の働きを助ける為のフレームリンクユニット (Frame Link Unit - FLU)

の4つの部分からなる。

まず、学習者の検索要求が TML を介して受けつけられ、FLU に基づく質問応答によって、学習者は、より適切な概念の指定へ導かれる。

システムは、この指定に基づき、 KDU と FLU を用いて関連した知識単位を探索するが、この際、コンテキストと、知識のカテゴリを選ばせることによる絞り込みを行う。

こうして KDU が選ばれると、該当するテキストの提示が行なわれる。また、その知識単位を、より基礎的な KDU の連鎖によって証明ないし説明するために、関連 KDU が呼び出すテキストを、一部変更や图形の置き換えをしながら接続して提示することができる。

学習者は、テキストと同時に表示される関連 KDU 名にポイントすることで、このような表示を制御できる。

さらに、このような表示の制御は、いくらでもネスティングしてゆくことができ、学習者は、知識の森の奥深くに分け入ることができる。システムは、このルートをスタックしておき、これによって学習者をもとの KDU にまで呼び戻すことができる。

学習者がシステムには入っていない問題の解決を行なおうとするときには、たとえば“こんなコンテキストで、こんな結果を生ずる原因を知りたい”といった検索要求に応えることができ、そのとき、学習者が

指定するコンテキストに合わせて、テキストの表示をmodifyする。その後は、関連したK D Uのメニューのなかから選ぶことによって対応したmodificationが、該当するテキストにも施されて表示が進行することにより、鎖の輪の探索を援ける。

3. 知識の表現

すべての記述の枠組みとなるCFUについて図1に例をしめす。

CFU0 レールの圧延 (self, ...)

CFU1 鋼塊の圧延 (self, 鋼塊 (, 炭素含有率), 製品 (, 用途), 圧延システム)

CFU2 鋼塊 (self, 炭素含有率)

CFU3 工業製品 (self, 用途)

CFU4 符号化方式 (self [メッセージ, 符号系列])

CFU5 群 (self [要素], 演算)

CFU6 部分と全体 (全体, 部分 {})

CFU7 白い (色が)

図 1.

CFUの各々は、物 (CFU2,3) , 行為、操作 (CFU0,1) , 事象、系 (CFU4,5) など、ある概念とその属性の組み合わせについての情報を与えるので、その概念があてはまる個々の対象を値としてとる基底スロット (selfとする) と、一般に複数の属性に対する属性スロットを持つ。

このほか、さまざまな関係を表わすCFU (CFU6,7) があり、これらには基底スロットは設けない。

さて、CFUどうしの関係については、例えば、CFU0の基底スロットの要素はCFU1の基底スロットの要素でもある。この場合、CFU1は、CFU0の上位フレームであるという。また、同様の意味で、CFU1の製品スロットは、CFU3をその上位フレームとする。こんな場合、CFU1の製品スロットにCFU3の全スロットを引き込むことが許される。こうし

て得られたCFU1のすべての属性スロットをCFU0のコンテキストの限定の下に（例えば、製品は、レールになる）その属性スロットとして置くことができる。こうした属性の引用を遡行ということにする。遡行によって得られた属性については、そのオリジナルの属性に関する記述 (FLU) が継承できる。

CFUの基底スロットがとることのできる値としては、当該CFU概念の外延要素としてそれ以上識別する必要のない最小単位の要素を想定する。それは、上位CFUのあるスロットの外延を、他のCFUのスロットの外延との共通部分に縮小することにより、基底スロットが限定され下位CFUが得られるが、こうしてついに单一の要素にまで限定することを原理的に可能にしておかないと、階層関係に破綻を来すおそれがあるからである。

この仮定から、属性スロットの値は、基底スロットの値に対して、一価の関数依存関係であるべきであることが帰結する。実際、仮に、基底スロットの1つの値に対して、ある属性のとる値が二つあったとする。もしこの値の差が、基底スロットの値としての要素を細分化するのに役立つのであれば、基底スロットの値は、最小単位の要素ではなかったことになる。値の差がそのような細分化に役立たないなら、属性値としても区別する必要はなく、1まとめにして扱えばよい。

つぎに、FLUは、CFU間の諸関係、特に階層的あるいは構造的関係を、CFUを一階述語として引用した含意形式で記述する。FLUの例を図2に掲げる。

FLU1. 雷鳥 (self *P)

=> 鳥が生きる (鳥 *P, 時 *T, 羽毛の色

*FC), 冬 (*Tが), 白い (*FC が) FLU2. コンピュータでAV機器を制御する

(self*C, コンピュータ*C(, オペレーティングソフト*S), AV機器*A(, メーカ*M))

=>制御する(, *Cが, *Aを), コンピュータ(self*C, オペレーティングソフト*S), AV機器(self*A, メーカ*M)

図 2.

このように, FLUは概念の上下関係を陽に記述するので, FLUを登録する度に, 先に述べた巡回によるCFUスロットの設定を示唆し, これに基づいて他のFLUとの間の関係を見直すことにより, これらの知識ベースの編集の矛盾をチェックすることができる。

KDUは, 1つのまとまった内容を持つ知識の単位について, その内容の記述を含むテキストへのポインタを持ち, 他方, 学習者がこれを検索する為の手掛けりを与え, 自らテキストの提示の制御を行なわせるためのインターフェースとなる知識表現をもつ。

知識のカテゴリに対応して, 次のようなカテゴリの KDUが必要であると思われる。

- (1) 概念フレーム記述 (framedesc)
- (2) 論理的言明 (assertion)
- (3) 論理的関数関係 (function)
- (4) 因果/操作的言明 (causality)
- (5) 因果/操作的関数関係 (causfunc)
- (6) 関数評価手続き (evaluation)

それぞれの場合について図3以降にその例を示すが, KDUの表現にはつきのような要素が必要である。

KDU <id-KDU> <title>
category: <category name>
context : <CFU with variables>, <
<role slot1>: <CFU with var.+>, <
<role slot2>: <CFU with var.+>, <
.....
- see TEXT <id-TEXT>
refer-to (<id-KDU with instantiated

variables>, <...>)

referred-to (<id-KDU>, <...>)

+ function, causfunc, evaluation categoryのときは, <variables>

role slot's は, 各categoryに対して, framedesc: なし

assertion: 前提 premise, 結論

conclus

causality: 因状態 caustate, 因事象 cauevent, 果事象 effevent, 果状態 effstate

function/causfunc/evaluation: 関係 related, または, 与件 given, 結果 result

context は, 知識の適用範囲を与えることと検索を絞り込むことの二つの働きをする。

検索の支援には, context と, 各role slot をキーとして用い, FLUを参照してその階層を辿ることにより, 各知識の適用可能性を調べる。

refer-toは, 当該 KDUが参照する KDUの変数をインスタンス化し, 相互に接続したグラフであって当該 KDUの内容の解説/証明/問題解決の筋道に対応しており, これにより参照テキストの制御を起動する。referred-to は, 当該 KDUを参照している KDUのリストであって, 逆向きの参照の便宜を図る。

テキストの検索と制御をすべてテキストからは独立した KDUを通して行っているのは, テキスト自体は, 頻繁に改定されるものなので, これを分離した方がよいこと, テキストには, その了解性をたもつために, いくつかの知識単位から参照されるような記述を許すべきであることによる。

4. 検索要求を誘導する為の質問応答について

学習者が TMLを介して行う検索要求の表現は, 学習者自身の要求を的確に反映し

ているとは考え難い。そこで、システムは、学習者を促して、自分が表現した概念が、システムのなかで、どのような意味付けを持たされているのか、なんらかの意味で関連する他の概念にはどんなものがあるのかについて質問をさせ、その答えを知った学習者がより的確な表現に行き着くことを期待する。

そのような質問のうち、指定した概念に属する例（下位概念）を示すこと、および指定した概念を含む（上位の）概念を示すことは、最も基本的な意味付けと連関を与えるものである。これは、FLUを用いて実施することが可能である。例示を求める場合、概念の指定は、単一のCFUのスロットで該当するものを知らないときには、複数のCFUのスロットを用いて行わざるをえない。また、的確な表現を思いつかないが、いくつかの概念をその例として持つような概念（類比の根）を求めたいこともある。次に示すのは、そのような例示あるいは類比の根を求める質間に答えることのできるシステムを、VAX11/780 上のPROLOG/KR を用いて試作した結果である。

システムはFLUの形をした知識として、

家庭教師は大学生であり、教える主体である；大学生は、教えられる対象であり、教えられる場所は大学である；高校生は、教えられる対象であり、教えられる場所は高校である；大学、高校は学校である。

を持っている。これらの知識は前件が单一の、後件が複数のリテラルからなるが、スコーレム変数を導入して複数の $A \rightarrow B$ なる形のホーン節に書き直す。

(get A B C) を、 $D \rightarrow A \wedge B \wedge C$ なるDつまりA,B,C の例示を見つける述語とする。

このアルゴリズムは次のように書ける。

(get A B C) を実行すると、

- (1) $D \rightarrow A$ なる Dをとってくる。
- (2) 述語 Dに対応するある特定のFACT D^+ を assertする。
- (3) D^+ から $B \wedge C$ が証明されれば、 Dを解とする。失敗すれば、 D^+ を retractして、(1) にバックトラックする。
- (4) (1)-(3) でのバックトラックの可能性がついた場合、 $D \rightarrow A$ なる Dをとってきて、(get D B C) を実行する。

(1)-(3) の過程はbreadth-first の探索であるが、(4) で再帰的にget を呼んでいるため、これ以降はdepth-first になっている。 D^+ は、D に現われる全ての変数をある定数に置き換え、スコーレム変数、ないし定数を、ある変数に置き換えたものである。

類比の根を求める場合も、 $(A \rightarrow D) \wedge (B \rightarrow D) \wedge (C \rightarrow D)$ と $\sim D \rightarrow \sim A \wedge \sim B \wedge \sim C$ が同値であることを利用すれば、 $\sim A \sim B \sim C$ に関する具体例の提示に置き換えられ、やはりget により実行できる。ただし、否定を取り扱うため、述語の第一引数に+/- のどちらかを置き、肯定、否定に対応させる。また、ルールはその対偶も登録するものとする。

以下に質問応答例を示す。* で始まるのは変数であり、(sc ...)の形をしているのがスコーレム変数である。

```
: (get (teach + (sub *x)(obj *y)(place *z))
      :   (teach + (sub *u)(obj *x)(place *w))
      :   (school + (self *w)))
      ; 教える主体であり、かつ対象でもあり、
      ; 教えられる場所が学校であるものは?;
      if (private-teacher + (self g00693))
        (get (teach + (sub g00693)
                  (obj (sc 8 g00693)))
```

```

        (place (sc 9 g00693)))
(teach + (sub (sc 6 g00693))
            (obj g00693)
            (place (sc 7 g00693)))
(school + (self (sc 7 g00693)))))

; 家庭教師であればよい;

:(get (highschool-student - (self *x))

:      (univ-student - (self *x)))
; 高校生と大学生の共通点は？（高校生でなく、大学生でないものは？）;

if (teach - (sub *g00743) (obj g00764)
                (place *g007
78))
(get (highschool-student - (self g0076
4))
      (univ-student - (self g00764)))
; 教育の対象である点である。（教育の対象でなければ、高校生でなく、大学生でもない）;

```

5. 今後の課題

テキストの提示を制御するアルゴリズムなどまだ不確定な部分が多い。今後、さらに仕様を具体化して試作へつなげたい。

参考文献

- (1) D.G.Bobrow and T.Winograd :"An Overview of KRL", Cognitive Science , vol.5,no.4,Oct-Dec,1981

K D U - 0 1 なぜ雨が降るか？（地理的関係1001）category:causality
context : 地理的関係1001（海域*PW, 山*M, 海岸部*C, *UC, *UI, 内陸部*I）,

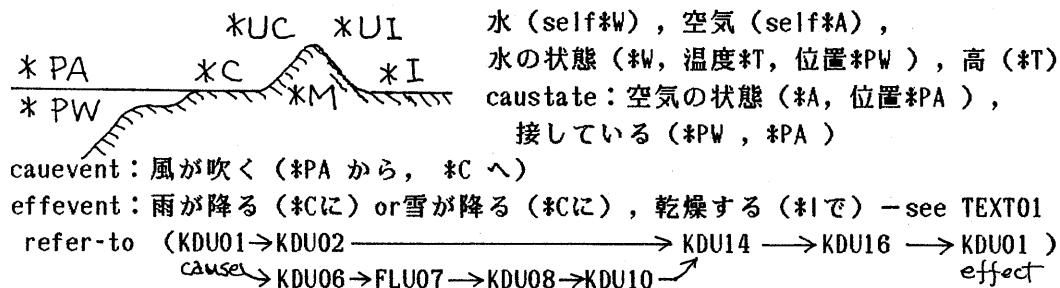


図 3 - 1

- (2) N.J.Nilsson : Principles of Artificial Intelligence ,Springer,1982
- (3) G.G.Hendrix:"Encoding Knowledge in Partitioned Networks " in N.Findler (ed.) Associative Networks pp.51-90, Academic Press,1979
- (4) R.C.Schank: Conceptual Information Processing ,North Holland,1975
- (5) J.H.Larkin:"A General Knowledge Representation for Learning or Teaching Science ", in A.C. Wilkinson (ed.), Classroom Computers and Cognitive Science ,Academic Press,1983
- (6) A.Stevens A.Collins S.E.Goldin Misconceptions in Students' Understanding", D.Sleeman & J.S.Brown(ed.), Intelligent Tutoring Systems ,pp.13-24 Academic Press,1980
- (7) C.D.Date: An Introduction to Database Systems, vol1,2 ,3rd ed., Addison-Wesley,1982
- (8) 堀, 斎藤, 猪瀬、"帰納的推論の記述に有用な知識表現の一提案"、情報処理学会論文誌、vol.24,no.1, PP.72-79, 1983
- (9) G. ポリア、数学の問題の発見的解き方、上、下、柴垣、金山訳、みすず書房、1964
- (10) 伊藤、" 学習支援システムの為の知識の構造化について"、信学技報 ET84-5
- (11) 伊藤、浅野、堀、横尾、" 学習支援システムにおける知識の検索および提示の為の知識表現について"、1985、信学総全大、S18-4

K D U - 0 2 水が蒸発する - category: causality

context : 水 (self*W), 空気 (self*A), 水の状態 (*W, 温度*T, 位置*PW)

caustate : 空気の状態 (*A, 温度*T, 位置*PA), 接している (*PW, *PA)

effevent : 水が蒸発する (水*W, *Aへ, 水蒸気*V)

effstate : 水蒸気が飽和している (*Aに, 水蒸気*V, 絶対湿度*HA),

空気の状態 (*A, 温度*T), - see TEXT02, referred-to (KDU01...)

K D U - 0 3 飽和絶対湿度は温度の増加関数 - category: causfunc

context : 空気 (self*A), 水蒸気が飽和している (*Aに, 水蒸気*V, 絶対湿度*HA),
空気の状態 (*A, 温度*T)

related : 飽和絶対湿度と温度 [given : 温度*T -> result: 絶対湿度*HA]

K D U - 0 6 地理的関係 1 0 0 1 で風が吹く - category: causality

context : 地理的関係 1 0 0 1 (海域*PW, 海岸部*C, *UC, *UI, 内陸部*I) -
空気 (self*A), 風が吹く (*PA から, *Cへ)

caustate : 空気の状態 (*A, 位置*PA), 接している (*PW, *PA)

effevent : 移動する (*Aが, *PA から, *Cへ) effstate : 空気の状態 (*A, 位置*C)

effevent : 上昇する (*Aが, *Cから, *UC へ) effstate : 空気の状態 (*A,
位置*UC)

F L U - 0 7 地理的関係 1 0 0 1 で地表とその上空

地理的関係 1 0 0 1 (海岸部*C, *UC, *UI, 内陸部*I)

=> 地表とその上空 (地表*C, 上空*UC), 地表とその上空 (地表*I, 上空*UI)

K D U - 0 8 空気は上昇すると断熱膨張する - category: causality

context : 空気 (self*A), 地表とその上空 (地表*E, 上空*U)

caustate : 空気の状態 (*A, 位置*E)

cauevent : 上昇する (*Aが, *Eから, *Uへ) effevent : 断熱膨張する (*Aが)

K D U - 1 0 空気は断熱膨張すると温度が下がる - category: causality

context : 空気 (self*A)

caustate : 空気の状態 (*A, 温度*T1) cauevent : 断熱膨張する (*Aが)

effevent : 温度が下がる (*T1 より, *T2 へ)

effstate : 空気の状態 (*A, 温度*T2)

K D U - 1 4 飽和水蒸気の凝結 - category: causality

context : 空気 (self*A), 水蒸気が飽和している (*Aに)

caustate : 水蒸気が飽和している (*Aに, 水蒸気*V1, 絶対湿度*H1),

空気の状態 (*A, 温度*T1)

cauevent : 温度が下がる (*T1 より, *T2 へ)

effevent : 水蒸気が凝結する (*Aから, 水蒸気*V1, 凝結水蒸気密度*CD, 凝結水*W,
残留水蒸気*V2), $H_2 = H_1 - CD$

effstate : 水蒸気が飽和している (*Aに, 水蒸気*V2, 絶対湿度*H2),

K D U - 1 6 雲ができる雨雪が降る - category: causality

context : 空気 (self*A), 地表とその上空 (地表*E, 上空*U)

caustate : 空気の状態 (*A, 位置*U)

cauevent : 水蒸気が凝結する (*Aから, 凝結水*W)

effevent : 雲ができる (雲*C, *Wから, *Uにて)

.....

四三-2

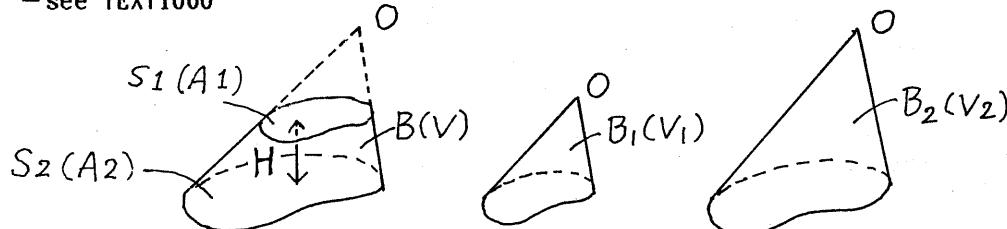
F L U 10

錐台 (self*B, 大錐体*B2, 切除錐体*B1, 上底面*S1, 下底面*S2,
 仮想頂点#0, 高さ#H, 上底面積*A1, 下底面積*A2, 体積*V)
 \Rightarrow 立体の部分と全体 (全体*B2, 部分{*B(体積*V), *B1}),
 錐体 (self*B1, 底面*S1 (面積*A1), 頂点#0,),
 錐体 (self*B2, 底面*S2 (面積*A2), 頂点#0,),
 2平面が相似の位置 (相似の中心#0, 相似な面1*S1, 相似な面2*S2,
 面の間の距離#H)

K D U 100 锥台 -category:framedesc

context 锥台 (self*B, 大錐体*B2, 切除錐体*B1, 上底面*S1, 下底面*S2,
 仮想頂点#0, 高さ#H, 上底面積*A1, 下底面積*A2, 体積*V)

-see TEXT1000



K D U 101 锥台の体積 -category: function

context : 锥台 (高さ#H, 上底面積*A1, 下底面積*A2, 体積*V)

related : [*H, *A1, *A2, *V]

-see TEXT1001

refer-to : (KDU101[H,A1,A2,V]-5<<5-KDU102[V2,V1,V] -3,4<<4-KDU103[H1,A1,V1]-
 0,1,2<<3-KD103[H2,A2,V2]-0,1,2<<2-KDU104[H1,H2,A1,A2]0<<1-KD105[H1,H2,H]-0<<0-KDU101)
 referred-to (KDU-”円錐台の体積” etc.)

K D U 102 立体の部分と全体 -category: function

context : 立体の部分と全体 (全体体積*V, 部分体積 {*V1 *V2})

related : [*V, *V1, *V2]

-see TEXT3002 , referred-to (KDU101etc...)



K D U 103 锥体 -category: function

context : 锥体 (体積*V, 底面積*A, 高さ#H)

related : [*H, *A, *V]

-see TEXT2003 , referred-to (KDU101 etc...)



K D U 104 2平面が相似の位置 -category :function

context : 2平面が相似の位置 (相似の中心#0, 相似な面1*S1, 相似な面2*S2, 相似
 中心からの距離1#H1, 相似中心からの距離2#H2, 面の間の距離#H)

related : [*H1, *H2, *A1, *A2]

related : [*H1, *H2, *H]

-see TEXT3004 , referred-to (KDU101 etc...)

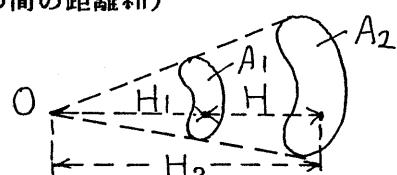


図 4.

K D U 500 平面上の3直線と垂線 - category:assertion

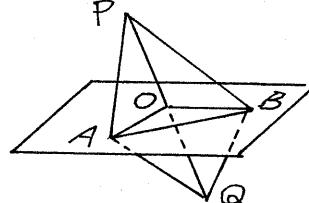
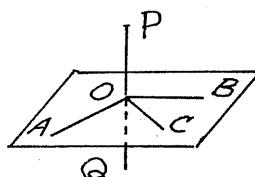
context : 幾何学配置5001(平面上の点 { *A, *B, *C } , 上の点 *P , 下の点 *Q , 交点 *O) ,

premise : 直交する (*OP , *OA) , 直交する (*OP , *OB)

conclus : 直交する (*OP , *OC) , - see TEXT5000

refer-to(KDU500 → FLU501 → KDU502 → KDU503 → KDU504 → KDU500)
 ↘ FLU501 → KDU502 ↗

F L U 501



幾何学配置5001(平面 *PL , 平面上の点 { *A, *B, *C, - } , 上の点 *P , 下の点 *Q , 交点 *O)

=> 平面上にある (平面 *PL , 直線 { *OA *OB *OC , - })

三角形 (頂点 (*A, *P, *Q)) , 点が線分上にある (線分 *PQ , 点 *O)

三角形 (頂点 (*B, *P, *Q)) , 三角形 (頂点 (*P, *A, *B)) , 三角形 (頂点 (*A, *B))

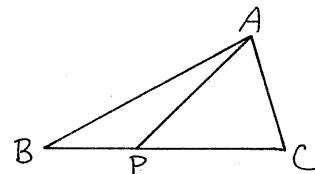
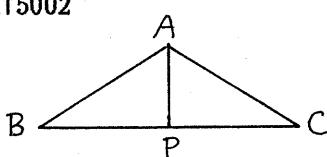
K D U 502 垂直二等分 - category:assertion

context : 三角形 (頂点 (*A, *B, *C)) , 点が線分上にある (線分 *BC , 点 *P)

premise : 垂直 (*AP , *BC) , *BP = *PC , conclus : *AB = *AC

premise : *AB = *AC , *BP = *PC , conclus : 垂直 (*AP , *BC)

- see TEXT5002

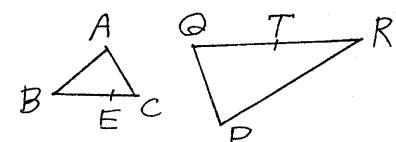
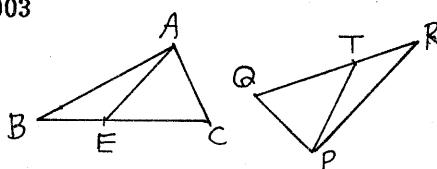


K D U 503 三角形の合同 - category:assertion

context : 三角形 (self*D , 頂点 (*A, *B, *C)) , 三角形 (self*S , 頂点 (*P, *Q, *R))

premise : *AB = *PQ , *BC = *QR , *CA = *RP , conclus : 合同 (*D, *S)

- see TEXT5003



K D U 504 合同と線分上の点 - category:assertion

context : 三角形 (self*D , 頂点 (*A, *B, *C)) , 三角形 (self*S , 頂点 (*P, *Q, *R)) , 点が線分上にある (線分 *BC , 点 *E) , 点が線分上にある (線分 *QR , 点 *T)

premise : 合同 (*D, *S) , *BE = *QT , conclus : *AE = *PT

- see TEXT5004