

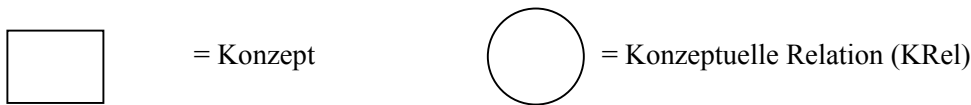
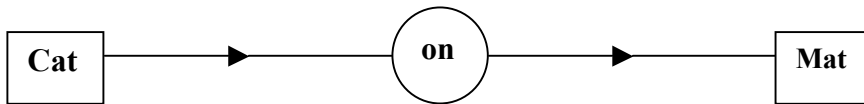
# Conceptual Graphs (CG)

## 1. Was sind CG's ?

Ein CG  $g$  ist ein zweiteiliger Graph der zwei verschiedene Knoten hat, die die Namen ‚Konzept‘ und ‚Konzeptuelle Relation‘ (KRel) tragen.

## 2. Syntax eines CG's

### 2.1. Graphische Darstellungsmittel



= Formel-Operator ( übersetzt ein CG in eine Prädikats-Kalkülformel): Kreise werden zu Prädikaten abgebildet, mit jedem Bogen als ein Argument, Konzepte werden zu getypten Variablen, wobei das Typ-Label innerhalb jedes Konzeptes den Typ bezeichnet. Wenn keine explizite Quantoren-Angabe gemacht wird, wird ein existentieller Quantor per default angenommen.

Beispiel: Natürlichsprachliche Repräsentation: The cat is on the mat.

Formel-Operator-Schreibweise:  $(\exists x : \text{Cat}) (\exists y : \text{Mat}) \text{on}(x,y).$

In linearer CG-Schreibweise :  $[\text{Cat}] \rightarrow (\text{On}) \rightarrow [\text{Mat}]$

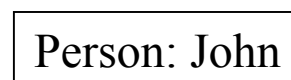
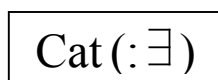
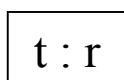
### 2.2. Wohlformtheitskriterien für einen CG

- a) Jeder Bogen  $a$  eines CG  $g$  muss eine KRel  $r$  in  $g$  mit einem Konzept  $c$  in  $g$  verbinden. In diesem Fall gehört der Bogen zur Relation  $r$  und der Bogen ist am Konzept  $c$  festgemacht, gehört aber nicht zu  $c$ .
- b) Ein CG  $g$  kann Konzepte enthalten, die mit keinen KRel verbunden sind. Aber jeder Bogen, der zu einer KRel in  $g$  gehört, muss genau an einem Konzept in  $g$  festgemacht werden.
- c) Drei verschiedene CG-Konfigurationen werden namentlich unterschieden:
  1. ‚blank CG‘ : Ein leerer CG ohne Konzepte, KRel oder Bögen.
  2. ‚singleton CG‘ : Ein CG, der nur aus einem einzigen Konzept besteht, ohne Bögen und KRel.
  3. ‚star CG‘ : Ein CG der aus einer einzigen KRel und den Konzepten besteht, die an den Bögen der KRel festgemacht sind.

### 2.3. Wohlformtheitskriterien für ein Konzept

Jedes Konzept hat einen Typ  $t$  und einen Referenten  $r$ , die nicht spezifiziert zu sein brauchen. Auf der Implementationsebene könnten dies Pointer-Paare sein, wobei der eine auf die Typ-Spezifikation zeigt und der andere auf die Referenten-Spezifikation.

Beispiele: Formal Konzept: „Cat“ Konzept: „Eine Person namens John“



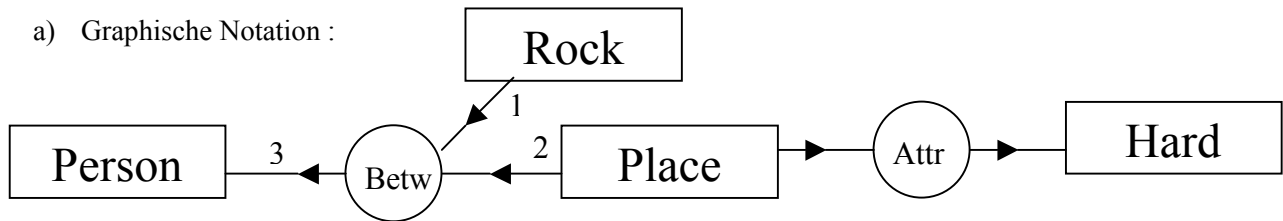
## 2.4. Wohlgeformtheitskriterien für eine KRel

- Jede KRel  $r$  hat einen Relationstyp  $t$  und einen nicht-negativen Integer  $n$ , Valenz genannt.
- Die Zahl der Bögen zu  $r = \text{Valenz } n$
- Eine KRel mit Valenz  $n$  heisst  $n$ -adisch, wobei die Bögen nummeriert sind  $(1, 2, \dots, n)$ .
- Für jede  $n$ -adische KRel  $r$  gibt es eine Sequenz von Konzept-Typen  $\langle t_1, t_2, \dots, t_n \rangle$ , die man als Signatur  $s$  von  $r$  bezeichnet.
- Eine 0-adische KRel hat keine Bögen und ihre Signatur  $s$  ist leer.
- Alle KRel des gleichen Typs  $t$  haben die gleiche Valenz  $n$  und die gleiche Signatur  $s$ .
- Es gelten überdies folgende Begriffe: monadisch: 1-adisch / dyadisch: 2-adisch / triadisch: 3-adisch

Die Signatur  $s$  ist gleichzeitig eine Selektionsbeschränkung in Bezug auf die Konzept-Typen, die an der KRel festgemacht werden könnten.

Beispiel : A person is between a rock and a hard place.

a) Graphische Notation :



b) Lineare CG-Schreibweise :

[Person]  $\leftarrow$  (Betw) –  
 $\leftarrow$  1 – [Rock]  
 $\leftarrow$  2 – [Place]  $\rightarrow$  (Attr)  $\rightarrow$  [Hard] .

## 2.5. Wohlgeformtheitskriterien für Lambda-Ausdrücke

- Für jeden Integer  $n$ , ist ein  $n$ -adischer Lambda-Ausdruck  $e$  ein CG, der Körper (body) von  $e$  genannt wird, in welchem  $n$ -Konzepte als formale Parameter von  $e$  benannt worden sind.
- Die formalen Parameter von  $e$  sind von 1 bis  $n$  nummeriert.
- Eine Sequenz  $\langle t_1, \dots, t_n \rangle$ , die Signatur von  $e$ , bei der jeder  $t_i$  der Konzept-Typ des  $i$ -ten formalen Parameters von  $e$  ist. Da ein 0-adischer Lambda-Ausdruck keine formalen Parameter hat, ist dessen Signatur die leere Sequenz  $\langle \rangle$ .

Beispiel: KRel. zwischen einer Person und einer Stadt:  $\langle \text{Person} \rangle$  is going to  $\langle \text{City} \rangle$  .

a) [Person : \_ 1]  $\leftarrow$  (Agnt)  $\leftarrow$  [Go]  $\rightarrow$  (Dest)  $\rightarrow$  [City: \_ 2]

## 2.6. Wohlgeformtheitskriterien für Konzept- und Relations-Typen

Stichwort	Konzept	Relation
Hierarchie	Eine Typen-Hierarchie ist ein partiell geordnetes Set $T$ , dessen Elemente Typen-Etiketten (type labels) genannt werden. Jedes Typen-Etikett ist als primitive Einheit (primitive) oder als definierte Einheit (defined) spezifiziert.	Jede Relationen-Hierarchie ist ein partiell geordnetes Set $R$ , dessen Elemente Relations-Etiketten (relation labels) genannt werden. Jedes Relations-Etikett ist als primitive Einheit (primitive) oder definierte Einheit (defined) spezifiziert.
Typ-Def.	Für alle Konzepte $c$ , ist der Typ von $c$ entweder ein Typen-Etikett in $T$ oder ein monadischer Lambda-Ausdruck.	Für jede $n$ -adische KRel $r$ ist der Typ von $r$ entweder eine Relations-Etikette in $R$ mit der Valenz $n$ oder ein $n$ -adischer Lambda-Ausdruck.
Grenzen	Die Typen-Hierarchie $T$ enthält zwei primitive Typen-Etiketten: der „Universale Typ“ : $\top$ der „Absurde Typ“ : $\perp$	Für jede Relations-Etikette in $R$ gibt es einen nicht-negativen Integer $n$ , der deren Valenz (valence) genannt wird.
Definition	Für jede definierte Typen-Etikette gibt es einen monadischen Lambda-Ausdruck, der seine Definition genannt wird.	Für jedes definierte Relations-Etikett mit der Valenz $n$ gibt es ein $n$ -adischer Lambda-Ausdruck der seine Definition genannt wird.

Austauschbarkeit	Beide sind gegenseitig auswechselbar und zwar in jeder Position, in der sie vorkommen könnten.	Ein definiertes Relations-Etikett und der entsprechende Lambda-Ausdruck sind <u>gegenseitig in jeder Position austauschbar</u> .
Ordnungsregeln	Die partielle Ordnung über T ist durch die Subtypen-Relation reguliert, mit den Symbolen $\leq$ für Subtyp (subtype), $<$ für Echten Subtyp (proper subtype), $\geq$ für Supertyp (supertype) und $>$ für Echten Supertyp (proper supertype). Wenn t irgendein Etikett ist, dann gilt $\top \geq t$ und $t \geq \perp$ und besonders $\top > \perp$ .	Die partielle Ordnung über R ist durch die Subtypen-Relation (subtype) reguliert, mit den Symbolen $\leq$ für Subtyp (subtype), $<$ für Echten Subtyp (proper subtype), $\geq$ für Supertyp (supertype) und $>$ für Echten Supertyp (proper supertype). Falls r ein n-adisches Relations-Etikett ist und s ein m-adisches Relations-Etikett, und n ist nicht gleich m, dann ist keines der nachfolgenden Aussagen wahr: $r < s$ , $r > s$ , $r = s$ .
Konsistenz	Die partielle Ordnung der Typen-Etiketten muss mit den Inferenzregeln, die über die Lambda-Ausdrücke definiert werden, konsistent sein.	Die partielle Ordnung der Typen-Etiketten muss mit den Inferenzregeln, die über die Lambda-Ausdrücke definiert werden, konsistent sein.

Beispiel: MaineFarmer = [Farmer: \_]  $\rightarrow$  (Loc)  $\rightarrow$  [State: Maine] .

- oder -

**type** MaineFarmer (\*x) **is** [Farmer: ?x]  $\rightarrow$  (Loc)  $\rightarrow$  [State: Maine].

## 2.7. Wohlgeformtheitskriterien für einen Referenten

a) Der Referent eines Konzeptes ist durch einen Quantor und einen Designator bestimmt.

b) Der Quantor kann eines der folgenden sein:

1. Existentieller Quantor: Dieser ist entweder durch das Symbol  $\exists$  oder durch die Abwesenheit eines anderen Quantorsymbols bestimmt. (existentieller Referent)

2. Definierter Quantor: Ein definierter Quantor ist ein Symbol oder ein Ausdruck einer erweiterten Syntax, der in ein CG übersetzt werden kann der nur existentielle Quantoren enthält.

c) Der Designator kann eines der folgenden sein:

1. Literal: Ein Literal ist eine syntaktische Repräsentation der Form eines Referenten. Es gibt Nummern, Zeichenketten und codierte Literale, die jeweils durch eine Paarangabe (Identifikator, Kette) bestimmt sind.

2. Locator: Ein Locator ist ein Symbol der festlegt, wie ein Referent gefunden werden kann. Die drei Arten von Locatoren unterscheiden sich in der Art, wie ein Referent definiert ist.

2.1. Individueller Marker: Bezeichnet ein einzigartiges Konzept im Individuen-Katalog einer Wissensbasis.

2.2. Indexikalischer Marker (Indexical): Ein Symbol, das den Referenten durch eine implementations-definierte Suche bezeichnet.

2.3. Name: Ein Symbol, das den Referenten unabhängig von der aktuellen Wissensbasis nach festen Konventionen definiert.

3. Deskriptor: Ein Deskriptor ist ein CG der den Referenten beschreibt. Ein ‚blank descriptor‘ repräsentiert einen ‚blank CG‘ als Deskriptor.

Der Referent eines Konzeptes bestimmt die Entität oder das Set von Entitäten auf die ein Konzept verweist. Der Designator spezifiziert den Referenten indem er seine Form (Literal) aufzeigt, auf ihn verweist (Locator), oder diesen beschreibt (Deskriptor). Ein existentieller Quantor postuliert die Existenz einer Instanz eines Typen. Ein definierter Quantor kann auch andere Quantitäten oder Masse postulieren.

Die definierten Quantoren enthalten auch den Universal-Quantor  $\forall$  den Quantitäts-Quantor @1, und Kollektionen, wie z.Bsp. {1,2,3} oder {Thomas, John, Frederik}.

## 2.8. Wohlgeformtheitskriterien für den Kontext

a) Ein Kontext C ist ein Konzept, dessen Designator ein nicht-leerer CG g ist.

b) Der CG g ist unmittelbar eingebettet in C (immediately nested) und jedes Konzept c von g ist ebenfalls unmittelbar eingebettet in C.

c) Ein Konzept c ist eingebettet in C entweder wenn c unmittelbar eingebettet ist in C oder wenn c unmittelbar in einem anderen Kontext D eingebettet ist, das seinerseits in C eingebettet ist.

d) Zwei Konzepte c und d sind gemeinsam eingebettet (co-nested), gdw. wenn entweder  $c=d$  oder wenn ein Context C existiert, in dem c und d unmittelbar eingebettet sind.

e) Wenn ein Konzept c ist gemeinsam eingebettet mit einem Kontext C, dann ist ein beliebiges Konzept d, das ebenfalls in C eingebettet ist, immer tiefer eingebettet als c.

f) Ein Konzept d ist innerhalb des Skopus eines Konzepts c, wenn entweder d mit c gemeinsam eingebettet ist, oder wenn d tiefer eingebettet ist als c.

## 2.9. Wohlgeformtheitskriterien für ein Ko-Referenzset

- Ein Koreferenz-Set C in einem CG g ist ein Set von Konzepten, die aus g ausgewählt wurden oder aus CG, die im Kontext von g eingebettet sind.
- Für jedes Koreferenz-Set C muss es ein oder mehrere Konzepte in C haben, die sogenannten dominanten Knoten von C, welche alle Konzepte von C innerhalb ihres Skopus beinhalten. Alle dominanten Knoten von C müssen gemeinsam eingebettet sein.
- Wenn ein Konzept c ein dominanter Knoten eines Koreferenz-Sets C ist, dann kann es nicht Teil eines anderen Koreferenz-Sets sein.
- Ein Konzept c kann Teil von mehr als einem Koreferenz-Set  $\{C_1, C_2, \dots\}$  sein, vorausgesetzt dass c nicht ein dominanter Knoten irgendeines Koreferenz-Sets ist.
- Ein Koreferenz-Set C kann aus einem einzigen Konzept c bestehen, der dann der domin. Knoten von C ist.

## 2.10. Wohlgeformtheitskriterien einer Wissensbasis (Knowledge Base)

Eine Wissensbasis ist ein Kontext vom Typ KnowledgeBase deren Designator ein CG mit 4 Konzepten ist.

1. Typen-Hierarchie (type hierarchy): Ein Kontext vom Typ TypeHierarchy dessen Designator ein CG T ist, der die partielle Ordnung von Typen-Etiketten und den monadischen Lambda-Ausdrücken für jedes definierte Typen-Etikett spezifiziert.
2. Relations-Hierarchie (relation hierarchy): Ein Kontext vom Typ RelationHierarchy dessen Designator ein CG R ist, der die partielle Ordnung von Relationen-Etiketten, deren Valenz und den Lambda-Ausdrücken für jedes definierte Relationen-Etikett spezifiziert.
3. Individuen-Katalog: (catalog of individuals): Ein Kontext vom Typ CatalogOfIndividuals dessen Designator ein CG C ist, der genau ein Konzept für jeden individuellen Marker enthält, der in irgendeinem Konzept der Wissensbasis auftaucht. Der Designator kann auch andere Konzepte und Relationen enthalten, die die Individuen näher beschreiben.
4. Extremster Kontext (outermost context): Ein Kontext vom Typ Assertion, dessen Designator ein CG O ist.

Die Inhalte der Wissensbasis müssen folgenden Bedingungen genügen:

- Jedes Typen-Etikett, das in jedem CG oder Lambda-Ausdruck in T, R, C und O benutzt wird, muss in T definiert sein.
- Jedes Relationen-Etikett, das in jedem CG oder Lambda-Ausdruck in T, R, C und O benutzt wird, muss in R definiert sein.
- Individuelle Marker in jedem CG oder Lambda-Ausdruck in T, R, C und O müssen in C definiert sein.

Die TypenLabels, die als MetaLabels fungieren, müssen nicht in T definiert sein. Wenn jedoch eine Wissensbasis Informationen über eine weitere Wissensbasis enthält, dann sollten diese Etiketten ebenfalls definiert werden. Der Extremste Kontext einer Wissensbasis ist das, was Peirce „the sheet of assertion“ genannt hat. Jeder CG, der unmittelbar in diesem extremsten Kontext eingebettet ist, enthält eine Proposition, die als wahr angenommen wird. Der Individuen-Katalog listet alle „named entities“ im aktuellen Diskurs-Universum auf.

Quellen und weiterführende Literatur:

HELBIG, HERMANN. Die semantische Struktur natürlicher Sprache. *Wissensrepräsentation mit MultiNet*. Berlin: Springer 2001. S. 335-363.

SOWA, JOHN F. Knowledge Representation. *Logical, Philosophical, and Computational Foundations*. Pacific Grove: Brooks/Cole 2000.

<http://users.bestweb.net/~sowa/cg/index.htm>

<http://users.bestweb.net/~sowa/krbook/index.htm>