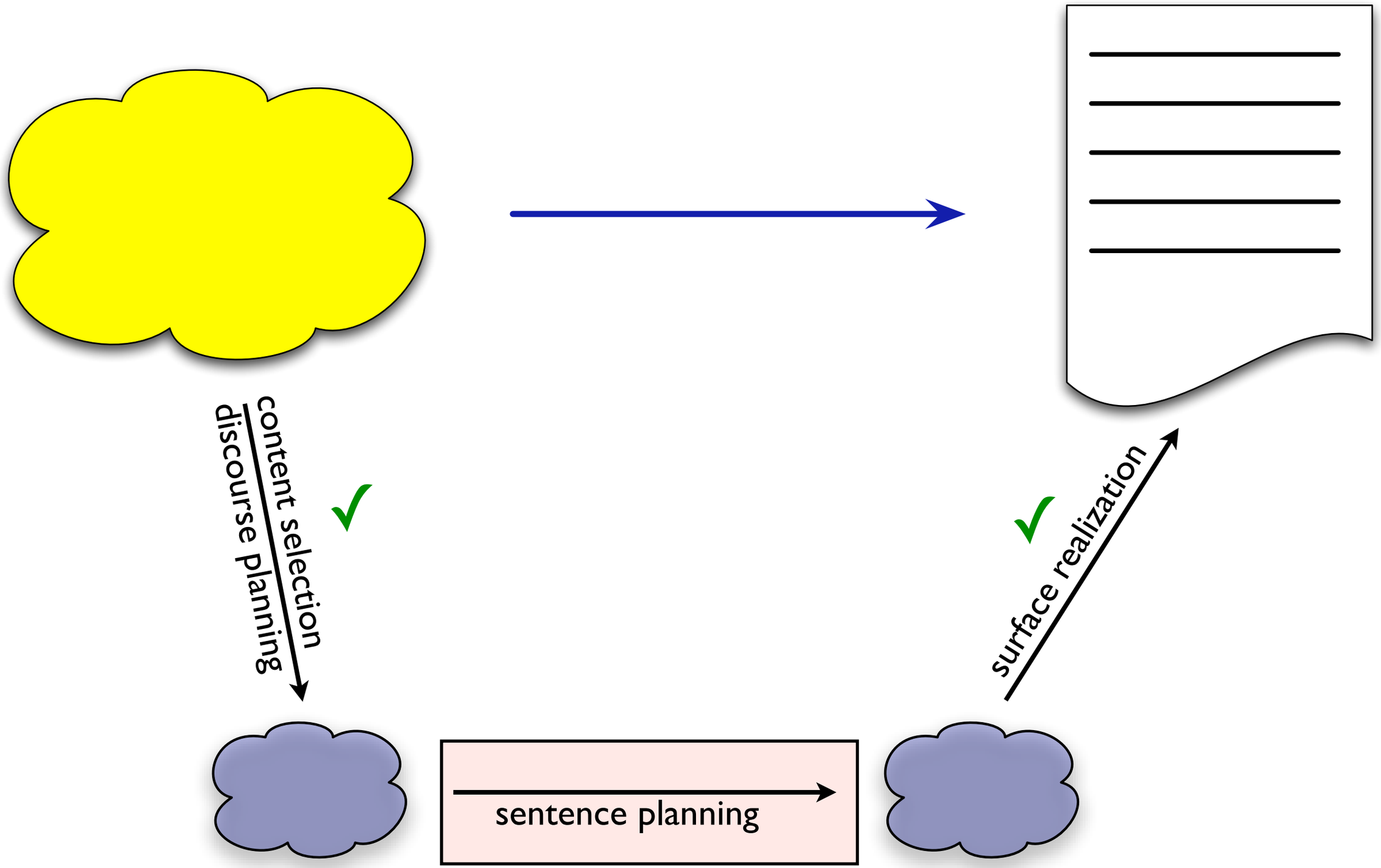
Referierende Ausdrücke

Proseminar “Generierung”

Alexander Koller

10. Dezember 2010

Wo stehen wir?

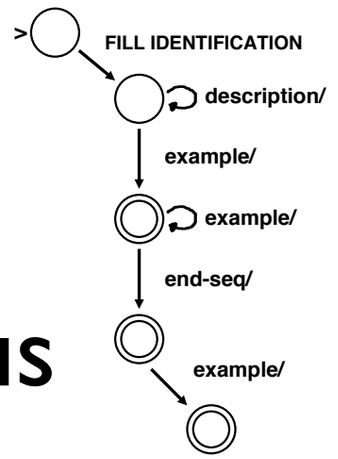


Die “Generation Gap”

a 

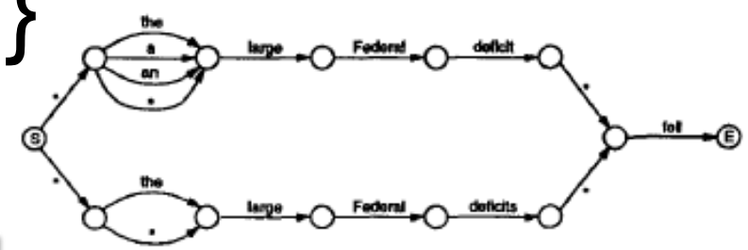
b 

Textplaner sagt:
drücke {sleeps(a)} aus



Woher kommt
{rabbit(a), white(a)}?

Realisierer: generiert Satz aus
{sleeps(a), rabbit(a), white(a)}



“Der weiße Hase schläft.”

Referierende Ausdrücke

- Ein charakteristisches Problem der Satzplanung: wie referiert man auf Objekte?

- Ziel: Hörer muss verstehen, welches Objekt in der Welt man meint.

- Sehr aktives Teilgebiet der NLG-Forschung.

Noch ein Beispiel

Umuz Ono, "The Attack of Hatchets"

A: 361, mostly goblins, 131 losses
D: 447 elves, 120 losses

Defender was victorious.

In 192, the human Ozud Oldstolen's left eye was torn out by the elf Avafi Matchedglossed the Assaulted Slaughters.
In 192, the elf Avafi Matchedglossed the Assaulted Slaughters's left foot was ripped by the human Ozud Oldstolen.
In 192, the elf Masami Glossrams's left upper arm was stabbed by the elf Arstruk Rabblevile.
In 192, the goblin Stâsost Flysoots was shot and killed by the elf Arane Pageford the Mighty Romance in Beancyclones.
In 192, the goblin Strodno Feverdoomed was shot and killed by the elf Gica Ownmusic the Enjoyable Sun of Venerating in Beancyclones.
In 192, the elf Ula Snakesperplexes was struck down by

ESC: done. Shift + ESC: back to age.

Noch ein Beispiel

Isard et al. (2003)

"This exhibit is a lekythos, created during the archaic period. It dates from circa 500 BC. It was painted by Amasis with the red figure technique and it originates from Attica."

"... Unlike the previous exhibit, it originates from Attica."

"Questo reperto è una lekythos."

"Αυτο′ το ε′κθεμα ει′ναι μια λη′κυθος."

Noch ein Beispiel

Dieser Vortrag

- Ansätze für RE-Generierung:
  - Dale & Reiter: inkrementeller Algorithmus
  - Krahmer et al.: graphbasierter Algorithmus
- Ansätze zur Evaluierung von RE-Systemen:
  - TUNA-Challenge

Attribute

- Dale & Reiter generieren *einfache* REs, d.h. eine Menge von *Attributen*.

- Attribute sind Eigenschaften von Objekten in der Welt, z.B.
  - Objekttypen: Knopf, Stuhl, Hase, ...
  - Farben: rot, blau, ...
  - Positionen: oben, links, ...

- REs werden als NPs realisiert, Attribute als Adjektive.

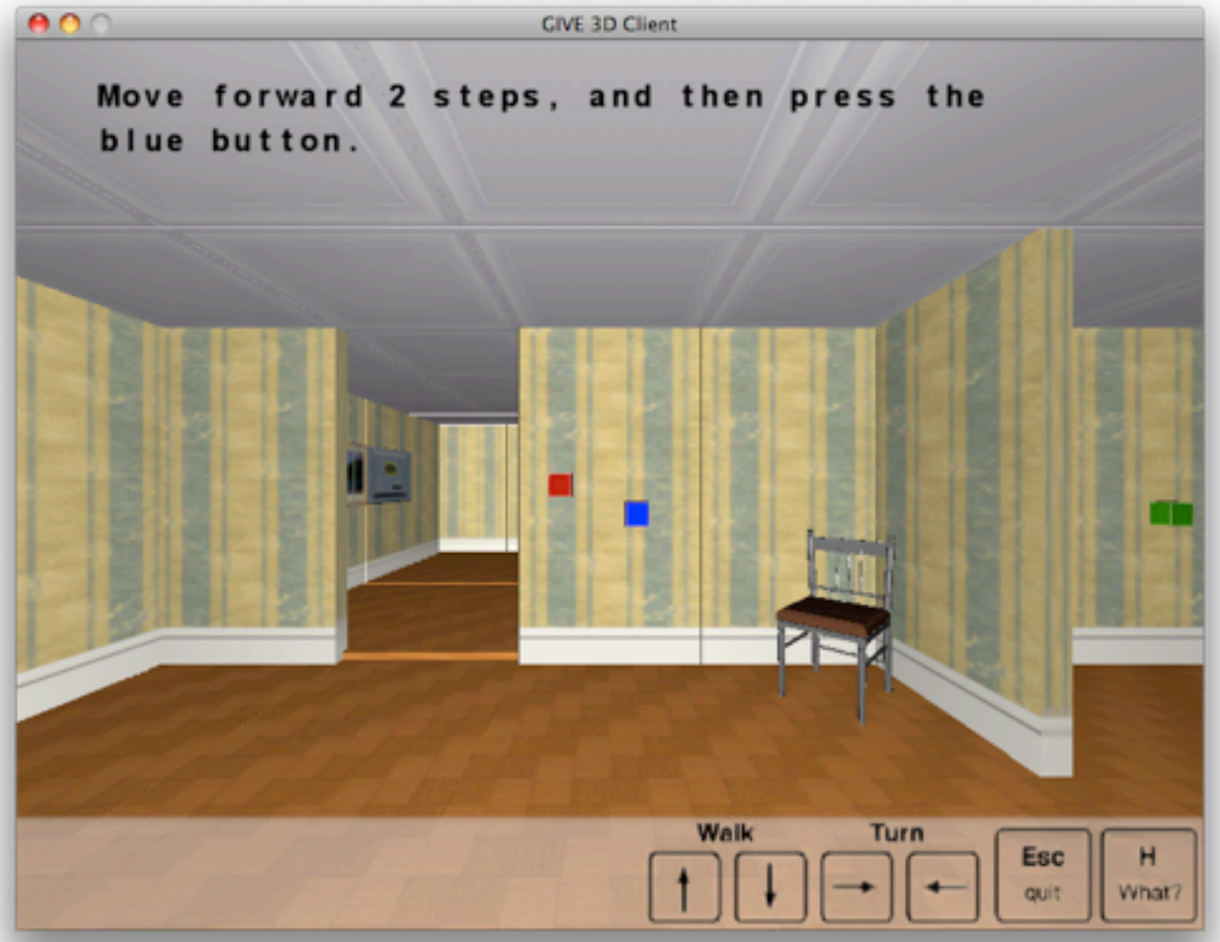
RE-Problem nach D&R

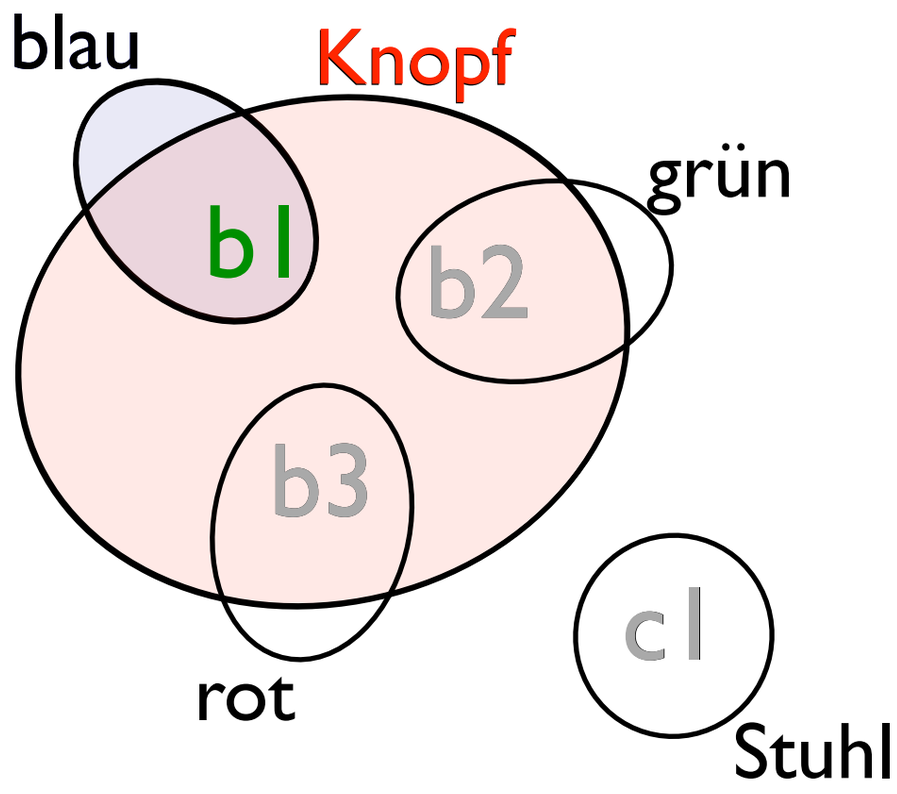
- Ein *referierender Ausdruck (RE)* für Zielobjekt t ist eine Menge S von Attributen, so dass
  - t alle Attribute in S hat
  - kein anderes Objekt im Universum alle Attribute in S hat

- REs referieren also *eindeutig* auf t.

- Andere Objekte, die die (unvollständige) Beschreibung S erfüllen, heißen *Distraktoren*.

Als Bild





grün: Zielobjekt; schwarz: Distraktoren

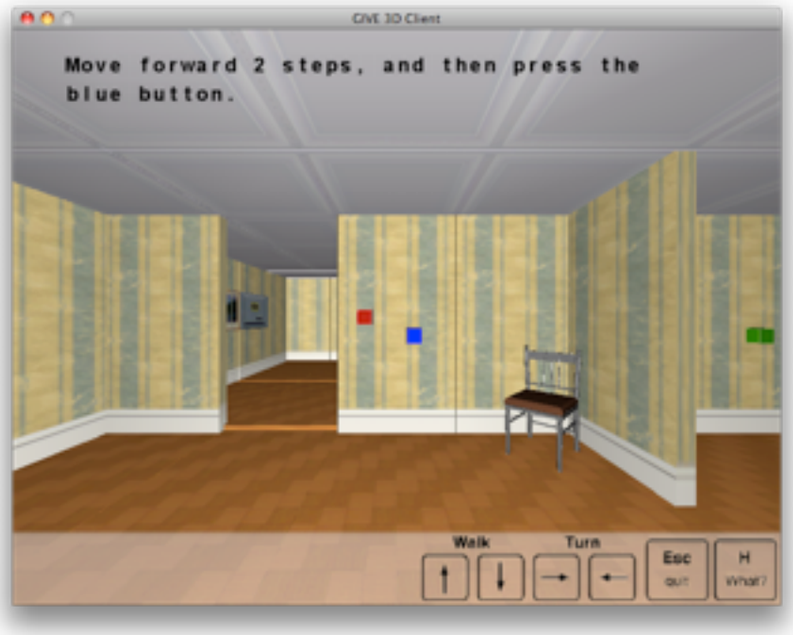
Wie berechnet man REs?

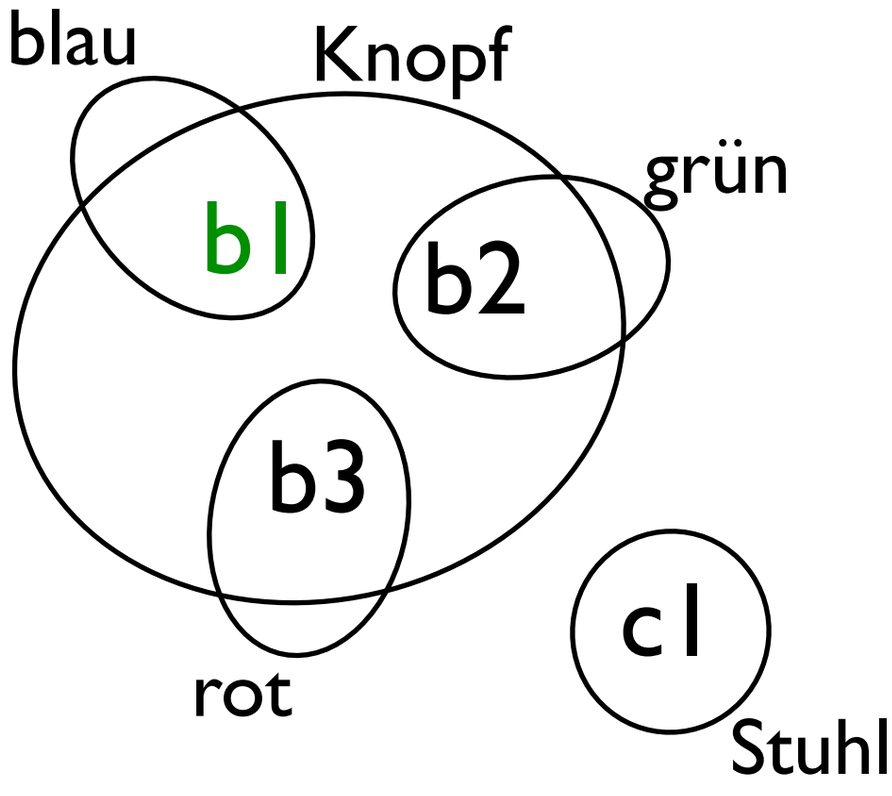
- Erster Ansatz: "Full Brevity"-Algorithmus (Dale 1992).

- Versuche, t mit einem einzigen Attribut zu beschreiben; dann mit allen Kombinationen von zwei Attributen; usw.

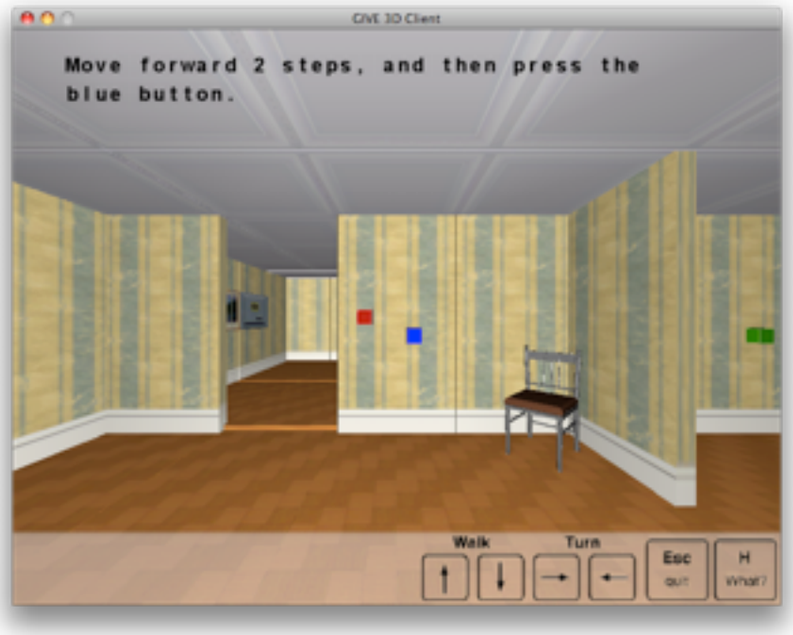
- Eindeutig referierenden Attributmenge gefunden ⇒ gib sie zurück.

Full Brevity

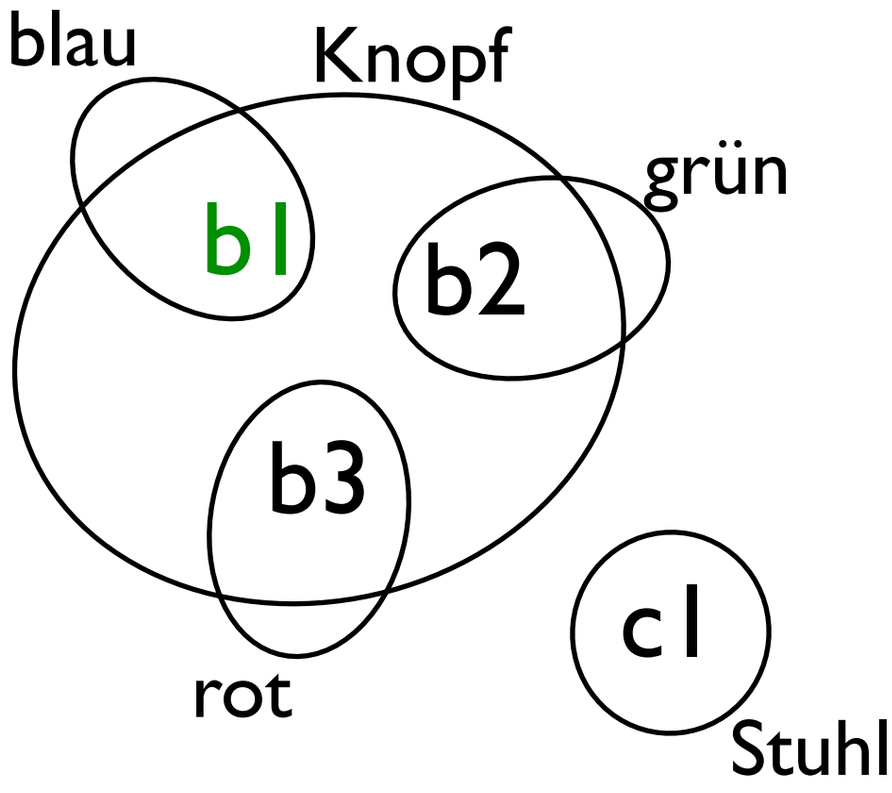




Full Brevity



ausgewählte Attribute:
{Knopf, blau}



Full Brevity: Diskussion

- Vorteil: Unter allen REs, die es gibt, berechnet FB-Algorithmus eine kürzeste.

- Nachteil: Berechnung von kürzesten REs ist NP-vollständiges Problem, d.h. es geht nicht effizient.

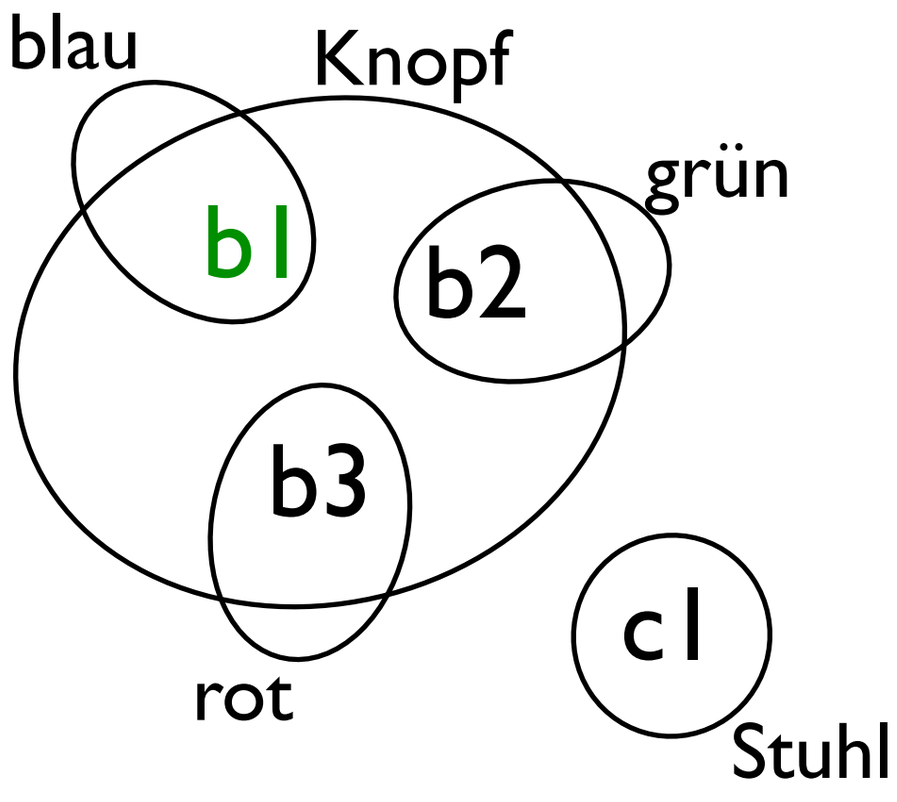
- Nachteil: Optimieren Menschen REs auf Länge?

Der inkrementelle Algorithmus

- Grundidee: Manche Typen von Attributen "kognitiv einfacher" als andere:
  - Objekttyp > Farbe > Größe > ....
- Inkrementeller Algorithmus:
  - betrachte Attribute der Reihe nach, sortiert von einfach nach schwierig
  - wenn t ein Attribut hat, füge es zur RE hinzu
  - wenn es keine Distraktoren mehr gibt, sind wir fertig
  - wenn uns vorher die Attribute ausgehen, gibt es keine RE

(Dale & Reiter 1995)

Der inkrementelle Algorithmus



Attribute:

- ▶ - Knopf ✓
- ▶ - Stuhl
- ▶ - blau ✓
- - grün
- - rot

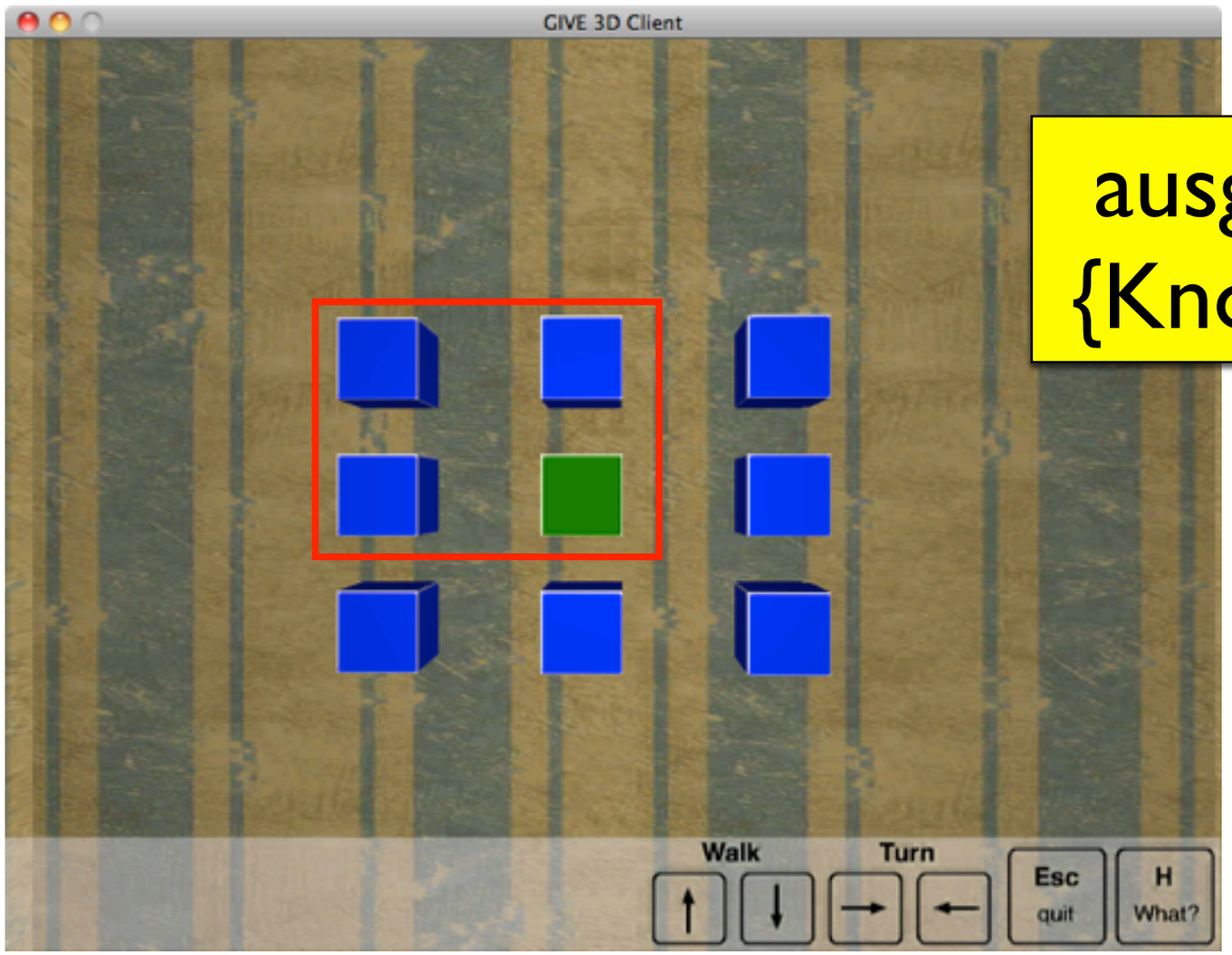
....

IA: Diskussion

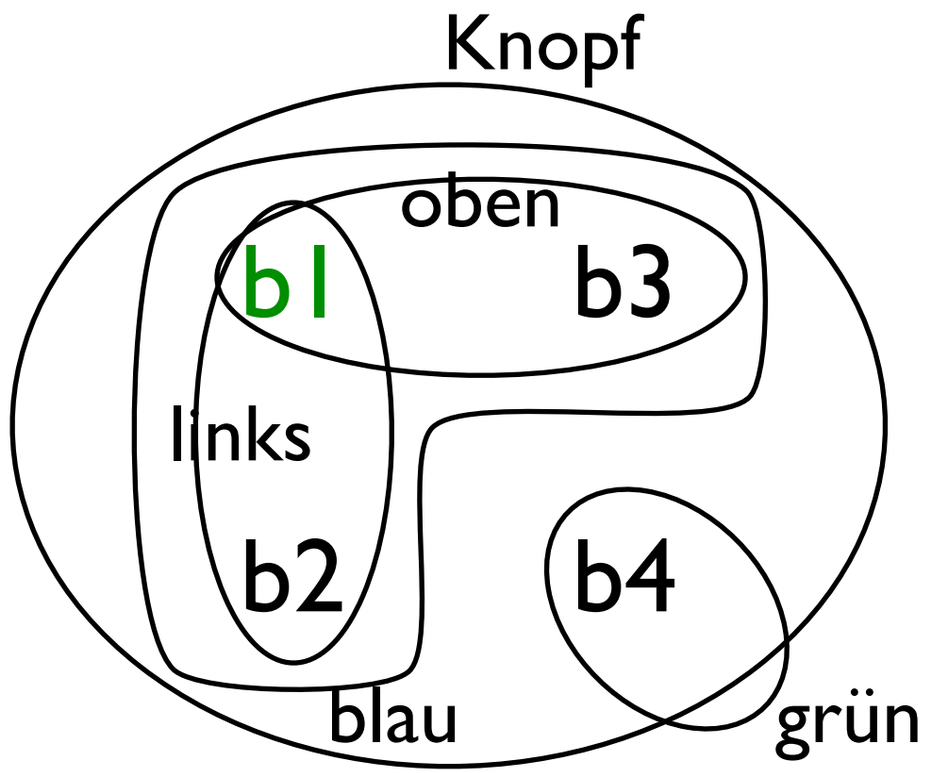
- Der inkrementelle Algorithmus ist *der* Standardalgorithms für RE-Generierung.

- Vorteile:
  - effizient
  - ganz einfach
  - kognitiv plausibel (?)

Nachteil 1: Unnatürliche REs



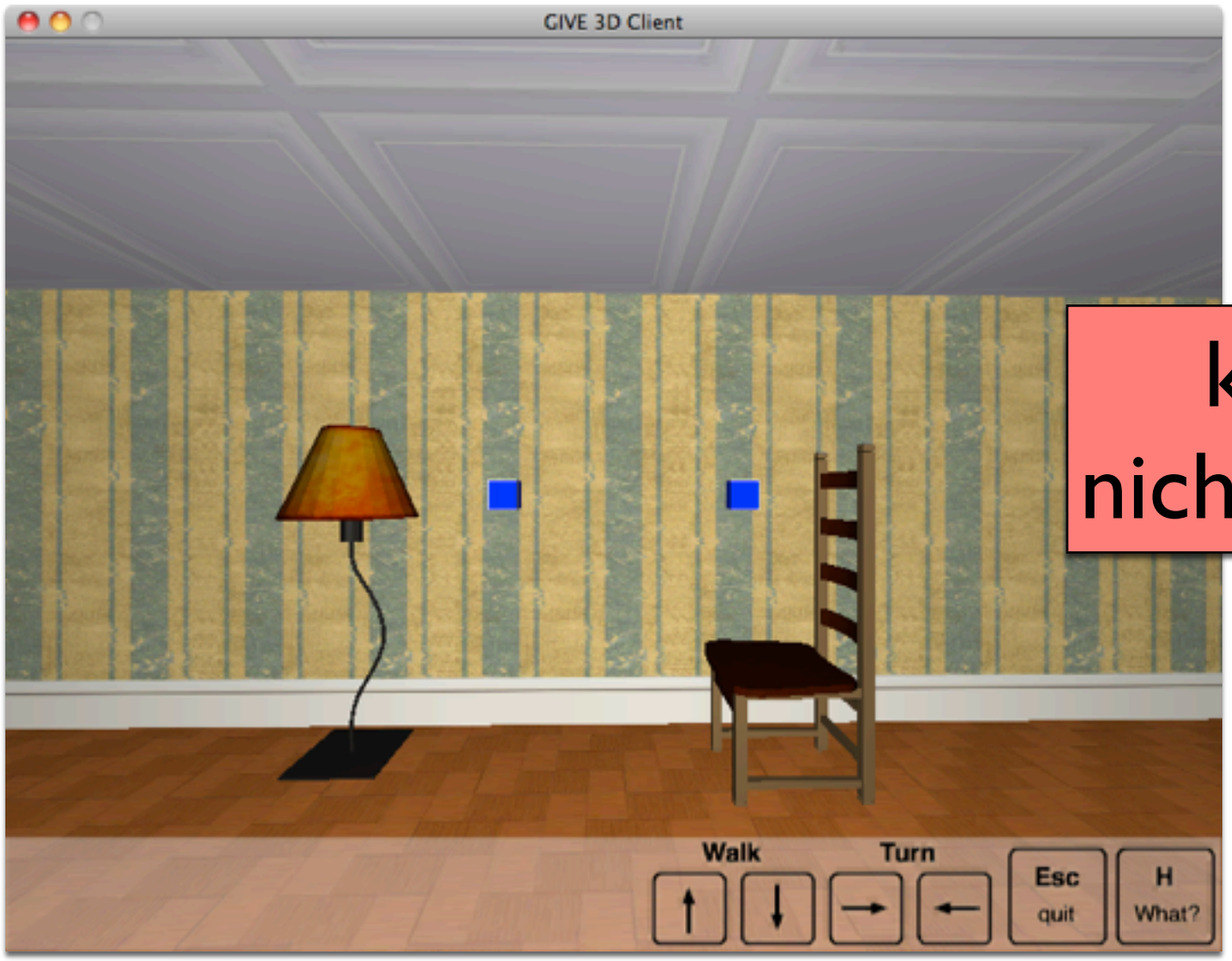
ausgewählte Attribute:
{Knopf, blau, links, oben}



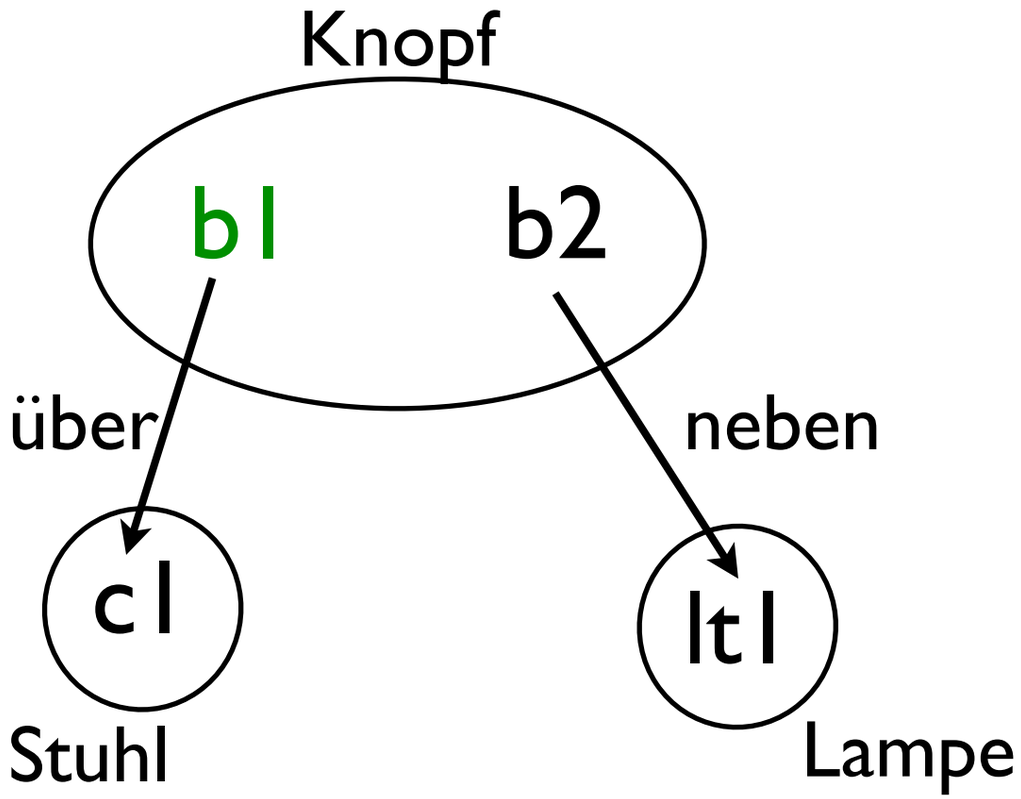
IA ist ein *greedy* Algorithmus

⇒ trifft manchmal suboptimale Entscheidungen

Nachteil 2: Relationale REs



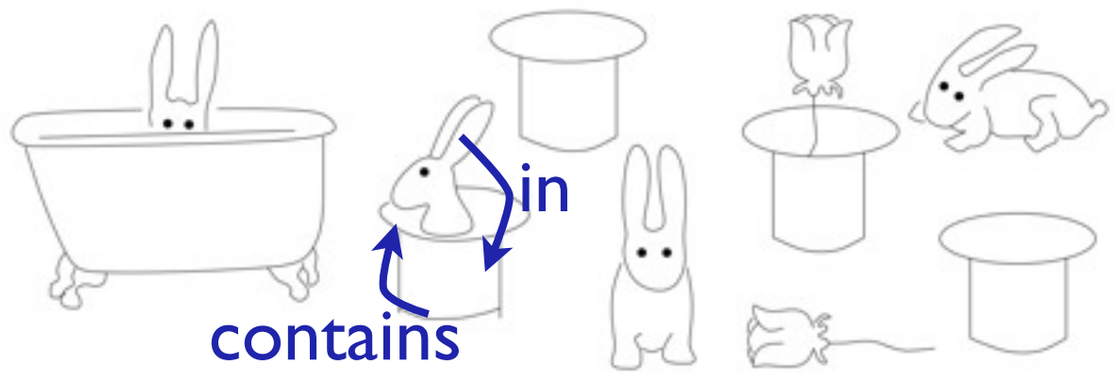
kann mit dem IA nicht generiert werden



Relationale REs

- Erweiterung von REs:
  - nicht nur Attribute
  - sondern erlaube auch *binäre Relationen* (neben, über, in, ...)

- Generierung von relationalen REs ist trickreiches Problem, z.B. muss man *unendliche Regression* vermeiden:

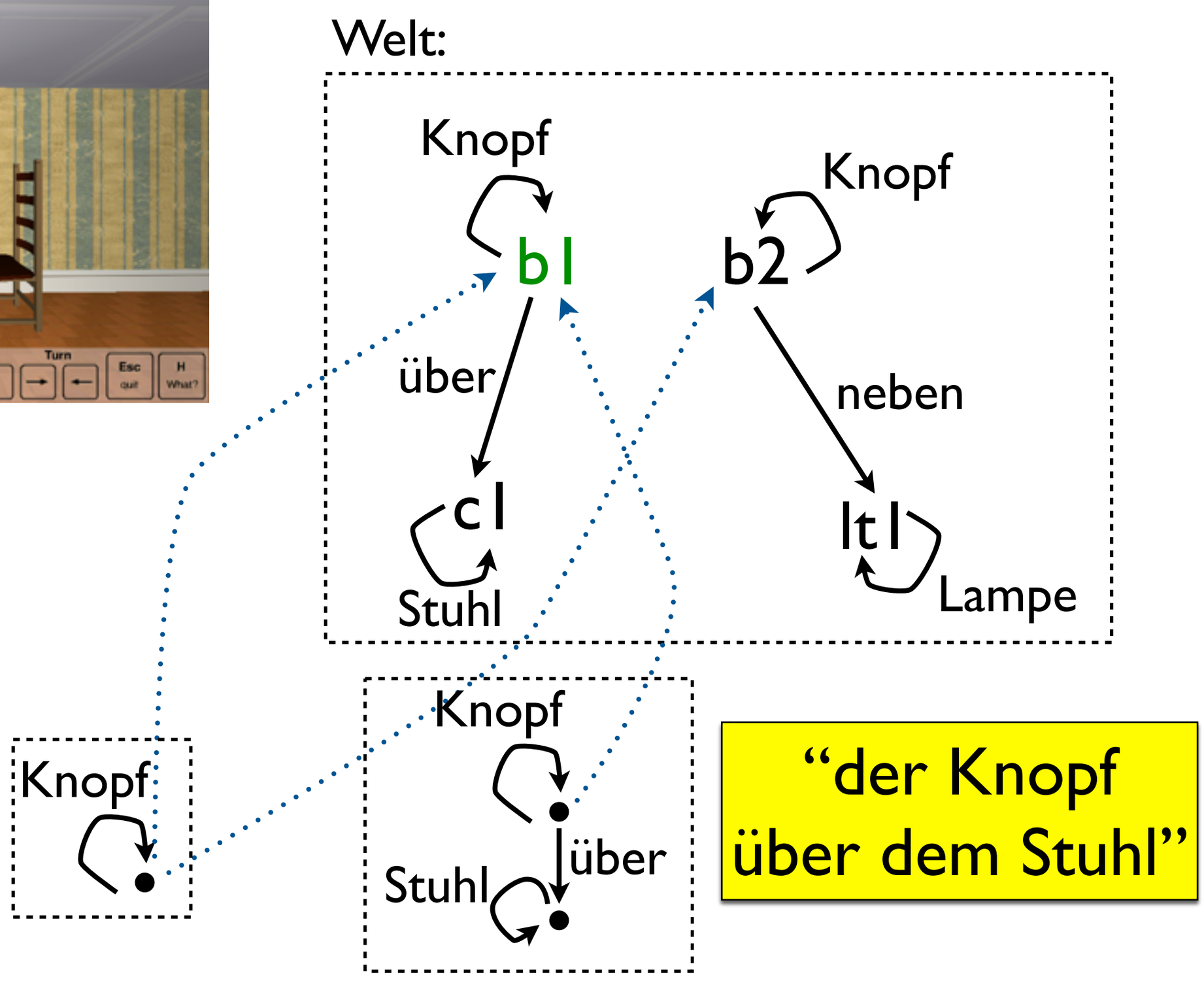


Krahmer et al.

- Krahmer et al. 03: Graphbasierter Algorithmus.

- Idee:
  - Welt und RE als gerichteten Graphen darstellen
  - Objekte sind Knoten
  - Attribute und Relationen sind Kanten
  - RE referiert eindeutig = RE-Graph kann nur an einer Stelle in Welt-Graphen eingebettet werden.

Ein Beispiel



Algorithmus (Grundstruktur)

```
findGraph(t, H) {
    wenn H nur bei t matcht, haben wir RE gefunden
    für jede zu H adjazente Kante e:
        findGraph(t, H+e)
}
```

REs für t berechnen: findGraph(t, [{t},∅])

Kosten

- Erweiterung des Ansatzes:
  - Kanten und Knoten haben *Kosten*
  - suche RE mit minimalen Kosten

- Algorithmus kann so erweitert werden, dass RE mit minimalen Kosten berechnet wird.

- Ersetzt Dale & Reiters Attributanordnung.

Komplexität

- Krahmers Algorithmus muss in jedem Schritt testen, wo H in den Graphen eingebettet werden kann
  = Subgraph-Isomorphismus-Problem.

- Dieses Problem ist als NP-vollständig bekannt.

- In vernünftigen Fragmenten kann Algorithmus trotzdem effizient sein.

Zusammenfassung, Teil 1

- Referierende Ausdrücke: eindeutige Beschreibung eines Objekts.

- Ansätze:
  - Dale & Reiter: Standardalgorithmus für einfache REs
  - Krahmer et al.: Algorithmus für relationale REs

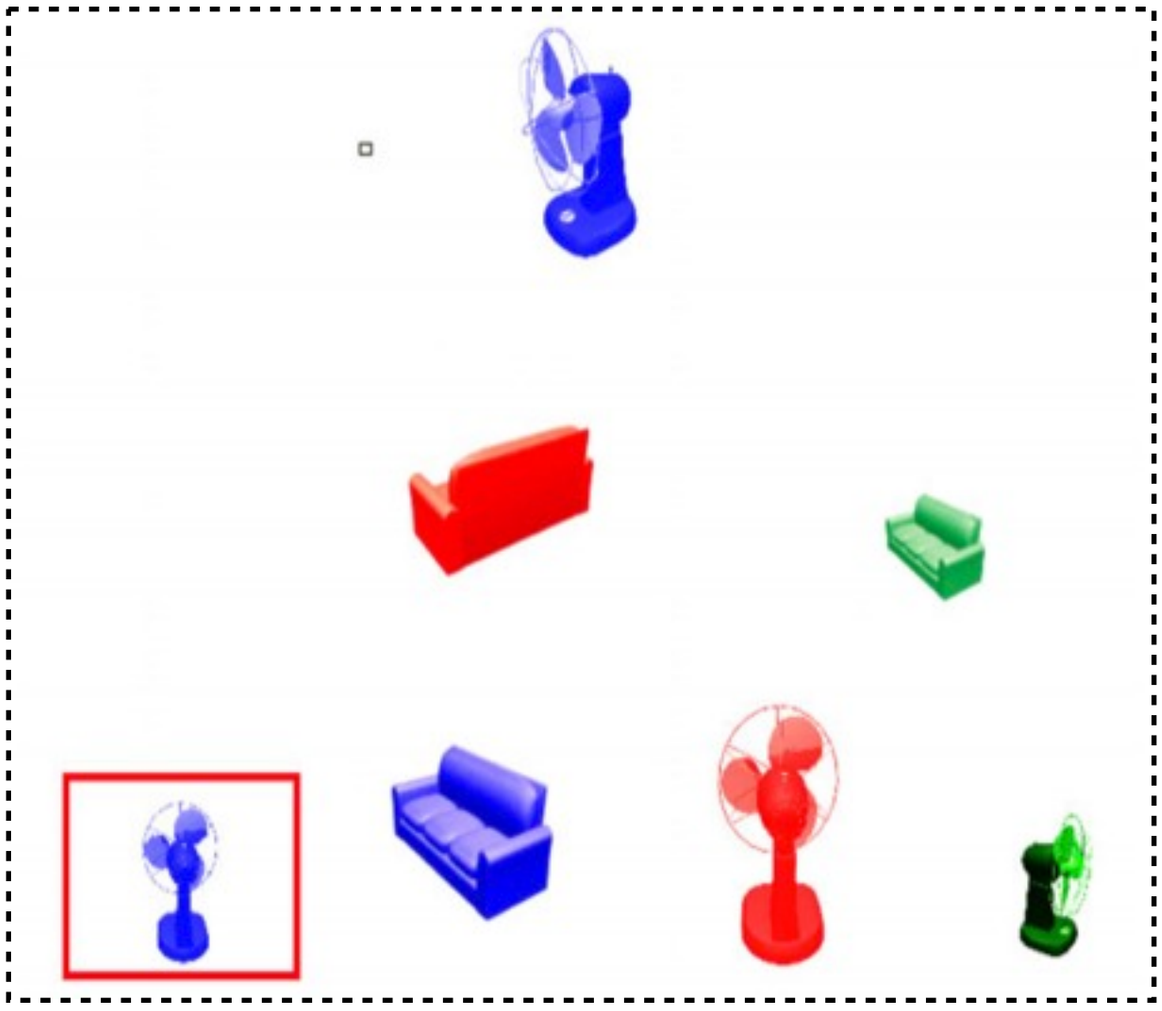
- Offene Frage: Woher weiß ich, welcher RE-Algorithmus besser ist?

Evaluierung von RE-Algorithmen

- Seit 2007: jährlicher Shared Task zur Evaluierung von RE-Algorithmen

- Ansatz:
  - Menschen annotieren REs für gegebene Szenen
  - Generierungssystem muss möglichst genau die annotierte RE wieder berechnen

Daten: TUNA-Korpus



Annotator 1 → "der kleine blaue Ventilator"

Annotator 2 → "der Ventilator links unten"

Evaluation

- NLG-Systeme erhalten Beschreibung der Szene als Input und generieren RE.
- RE wird mit annotierten REs verglichen:
  - Accuracy: Anteil exakt reproduzierter REs
  - BLEU: Überlappung von n-Grammen
  - String edit distance
  - DICE: Überlappung von ausgewählten Attributen

Ergebnisse

	All test data				People				Furniture			
	Acc	SE	BLEU	NIST	Acc	SE	BLEU	NIST	Acc	SE	BLEU	NIST
GRAPH	12.50	6.41	0.47	2.57	8.93	7.04	0.43	2.16	16.07	5.79	0.51	2.26
IS-FP-GT	3.57	6.74	0.28	0.75	3.57	7.04	0.37	0.94	3.57	6.45	0.13	0.36
NIL-UCM-EVOTAP	6.25	7.28	0.26	0.90	3.57	8.07	0.20	0.45	8.93	6.48	0.34	1.22
USP-EACH	7.14	7.59	0.27	1.33	0.00	9.04	0.11	0.46	14.29	6.14	0.41	2.28
NIL-UCM-VALUESCBR	2.68	7.71	0.27	1.69	3.57	8.07	0.23	0.94	1.79	7.34	0.28	1.99
NIL-UCM-EVOCBR	2.68	8.02	0.26	1.97	0.00	9.07	0.19	1.65	5.36	6.96	0.35	1.69
HUMAN-2	2.68	9.68	0.12	1.78	3.57	10.64	0.12	1.50	1.79	8.71	0.13	1.57
HUMAN-1	2.68	9.68	0.12	1.68	3.57	10.64	0.12	1.41	1.79	8.71	0.12	1.49

(TUNA Challenge 2009)

Vergleich der menschlichen Annotatoren miteinander

Fluency und Adequacy

- Zweite Klasse von Evaluationsmaßen: REs von frischen menschlichen Annotatoren bewertet.

- Zwei Maße:
  - Fluency: ist die RE vernünftiges Englisch?
  - Adequacy: beschreibt die RE das Zielobjekt richtig?

Ergebnisse

	All test data				People				Furniture			
	Adequacy		Fluency		Adequacy		Fluency		Adequacy		Fluency	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
GRAPH	84.11	21.07	85.81	17.52	85.30	18.10	87.70	14.42	82.91	23.78	83.93	20.11
USP-EACH	77.72	28.33	84.20	20.27	81.04	26.48	81.82	24.47	74.41	29.93	86.57	14.79
NIL-UCM-EvoTAP	76.16	28.34	61.95	26.13	78.66	27.48	59.13	29.78	73.66	29.22	64.77	21.79
HUMAN-2	74.63	34.77	73.38	27.63	80.93	31.83	73.16	30.88	68.34	36.68	73.59	24.23
NIL-UCM-ValuesCBR	72.34	33.93	59.41	33.94	68.18	37.37	46.23	34.92	76.50	29.86	72.59	27.43
HUMAN-1	70.38	34.92	71.52	30.79	83.39	24.27	72.39	28.55	57.36	39.08	70.64	33.13
NIL-UCM-EvoCBR	63.65	37.19	55.38	35.32	56.61	40.20	41.45	37.38	70.70	32.76	69.30	26.93
IS-FP-GT	59.46	40.94	66.21	30.97	88.79	19.26	65.27	32.22	30.14	35.51	67.16	29.94

(TUNA Challenge 2009)

Taskbasierte Evaluation

- Dritter Ansatz: frische Probanden bekommen Szene und RE gezeigt und müssen Objekt identifizieren.

- Zwei Maße:
  - identification accuracy: welcher Anteil von REs wurde korrekt aufgelöst?
  - identification speed: wie lange brauchen Probanden, um RE aufzulösen?

Ergebnisse

	All test data			People			Furniture		
	ID acc.	ID. speed		ID acc.	ID. speed		ID acc.	ID. speed	
	%	Mean	SD	%	Mean	SD	%	Mean	SD
GRAPH	0.96	3069.16	878.89	0.95	3081.01	767.62	0.96	3057.31	984.60
HUMAN-1	0.91	3517.58	1028.83	0.95	3323.76	764.59	0.88	3711.41	1214.55
USP-EACH	0.90	3067.16	821.00	0.86	3262.79	865.61	0.95	2871.53	730.15
NIL-UCM-EvoTAP	0.88	3159.41	910.65	0.88	3375.17	948.46	0.89	2943.65	824.17
NIL-UCM-ValuesCBR	0.87	3262.53	974.55	0.80	3447.50	1003.21	0.93	3077.56	916.87
HUMAN-2	0.83	3463.88	1001.29	0.89	3647.41	1045.95	0.77	3280.35	927.79
NIL-UCM-EvoCBR	0.81	3362.22	892.45	0.75	3779.64	831.91	0.88	2944.80	748.69
IS-FP-GT	0.68	3167.11	964.45	0.89	2980.30	750.78	0.46	3353.91	1114.68

(TUNA Challenge 2009)

Korrelationen

	Human-assessed, intrinsic		Extrinsic		Auto-assessed, intrinsic			
	Fluency	Adequacy	ID Acc.	ID Speed	Acc.	SE	BLEU	NIST
Fluency	1	0.68	0.50	-0.89*	.85*	-0.57	0.66	0.30
Adequacy	0.68	1	0.95**	-0.65	.83*	-0.29	0.60	0.48
Identification Accuracy	0.50	0.95**	1	-0.39	0.68	-0.01	0.49	0.60
Identification Speed	0.89*	-0.65	-0.39	1	-0.79	0.68	-0.51	0.06
Accuracy	0.85*	0.83*	0.68	-0.79	1.00	-0.68	.859*	0.49
SE	-0.57	-0.29	-0.01	0.68	-0.68	1	-0.75	-0.07
BLEU	0.66	0.60	0.49	-0.51	.86*	-0.75	1	0.71
NIST	0.30	0.48	0.60	0.06	0.49	-0.07	0.71	1

Lehren aus TUNA

- Beste NLG-Systeme generieren bessere (= verständlichere) REs als Menschen.

- Automatische Evaluierung von NLG-Systemen auf Grundlage von Korpora ist problematisch.

Aktuelle Forschungsrichtungen

- Grundlegende Frage: Was für REs soll ein NLG-System generieren?
- Psycholinguistische Ergebnisse: Menschen sind bei Produktion und Verstehen von REs relativ faul.
- Was macht REs verständlich?
- Realistischere Szenarien.

Schluss

- RE-Generierung ist aktives Teilgebiet von Satzplanung.
- Evaluations-Challenges für REs zeigen: In kleinen Domänen sind Systeme jetzt besser als Menschen.
- Generierung von verständlichen REs in komplexen Szenarien weiterhin interessant.