

# Einführung in die Computerlinguistik:

## Morphologie und Automaten I

WS 2011/2012

Manfred Pinkal

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## SHRDLU

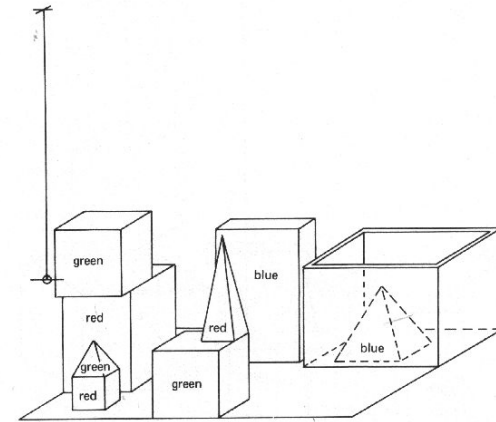
SHRDLU ist ein **wissensbasiertes Dialogsystem**.

Im Unterschied z.B. zu ELIZA verwendet es bei der Dialogführung unterschiedliche Arten von Wissen:

- Sprachliches/linguistisches Wissen
- Kontextwissen
- Weltwissen

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## SHRDLU: Ein wissensbasiertes Dialogsystem



Winograd's "Blocks World"

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Sprachliches Wissen in SHRDLU: Beispiele

### Morphologisches Wissen:

regelmäßige Verben bilden *grasp* ist regelmäßiges Verb  
Präteritum auf -ed *put* ist unregelm. Verb mit Prät. *put*

### Syntaktisches Wissen:

In Imperativen steht das *grasp* ist transitives Verb  
Verb an erster Stelle *stop* ist intransitives Verb

### Semantisches Wissen:

A+N in attributiven Konstruktionen bezeichnet *red* bezeichnet rote Objekte  
Dinge, die gleichzeitig unter *pyramid* ist Unterbegriff von *block*  
A und N fallen *grasp* bezeichnet eine Handlung, ...

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Sprachliches Wissen in SHRDLU: Beispiele

Grammatik	Lexikon
<p>regelmäßige Verben bilden Präteritum auf -ed</p> <p>In Imperativen steht das Verb an erster Stelle</p> <p>A+N in attributiven Konstruktionen bezeichnet Dinge, die unter A und unter N fallen</p>	<p><b>Morphologisches Wissen:</b> <i>grasp</i> ist regelmäßiges Verb <i>put</i> ist unregelm. Verb mit Prät. <i>put</i></p> <p><b>Syntaktisches Wissen:</b> <i>grasp</i> ist transitives Verb <i>stop</i> ist intransitives Verb</p> <p><b>Semantisches Wissen:</b> <i>red</i> bezeichnet rote Objekte <i>pyramid</i> ist Unterbegriff von <i>block</i> <i>grasp</i> bezeichnet eine Handlung, ...</p>

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Grammatisches und lexikalisches Wissen

- Morphologische, syntaktische, semantische Regularitäten sind tendenziell in der **Grammatik** kodiert
- Spezielle morphologische, syntaktische, semantische Information über Einzelwörter sind im **Lexikon** kodiert.
- Es gibt keine scharfe Grenze zwischen grammatischer Information und lexikalischer Information. Unterschiedliche linguistische Theorien schlagen eine unterschiedliche **Arbeitsteilung zwischen Grammatik und Lexikon** vor.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Außersprachliches Wissen

- **Kontextwissen:**
  - **Sprachlicher Kontext** / Dialoggeschichte: Welches Objekt wurden zuletzt erwähnt? (*Put it into the box.*)
  - **Situationskontext:** Welche Objekte kommen in der Äußerungssituation vor? (*What is the pyramid supported by?*)
- **Weltwissen:**
  - **Faktenwissen** ("Episodisches Wissen")  
"Es gibt zwei rote Klötze."  
"Die Kiste enthält eine Pyramide"
  - **Regelwissen:**  
"Zwei Objekte können nicht den gleichen Platz einnehmen."  
"Ein Objekt muss eine ebene Auflagefläche besitzen, damit ein zweites stabil darauf stehen kann"

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Wozu wird Wissen eingesetzt?

Wissen wird in der – menschlichen und maschinellen – Sprachverarbeitung eingesetzt, um Strukturen unterschiedlicher Arten und Ebenen aufeinander abzubilden:

- Speech → Text
- Text → Speech
- Wortkette → Bedeutungsinformation
- Bedeutungsinformation → Wortkette
- deutscher Satz → englischer Satz

Das zentrale Problem ist die **Mehrdeutigkeit (Ambiguität)** auf allen Ebenen: Wie kommen wir zu einer **eindeutigen Abbildung (Disambiguierung)**?

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Wie kommen wir an das Wissen?

### Zwei Optionen:

- **Manuelle Entwicklung:** Handkodierte Grammatiken, Lexika, Ontologien
  - Erlaubt die verlässliche und präzise Modellierung komplexer struktureller Zusammenhänge
  - Aber: Sehr aufwändig, Abdeckungsprobleme
- **Statistische Modellierung:** Anwendung von maschinellen Lernverfahren auf Sprachdaten
  - Effizienter Wissenserwerb, hohe Abdeckung, unterstützt flexible und robuste Sprachverarbeitung
  - Aber: Nur näherungsweise korrekt; die Brauchbarkeit nimmt mit zunehmender Komplexität der linguistischen Strukturen ab

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Teilbereiche der Morphologie

- **Flexion:**  
Deklination, Konjugation von Substantiven, Verben, Adjektiven, Pronomina  
*frag+te+st*  
*ge+frag+t*
- **Derivation** (Ableitung)  
*Er+kenn+ung*
- **Komposition** (Zusammensetzung)  
*Sprach+erkennung+s+technik*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Morphologie

- Morphologie ist der Teilbereich der Linguistik, der sich mit der internen Struktur von Wörtern befasst.
- Die wesentlichen Aufgaben der Morphologie in der Computerlinguistik sind
  - die Reduktion komplexer Wörter bzw. Wortformen auf ihre kleinsten bedeutungs-/funktionstragenden Bestandteile, die **Morpheme**
  - die Identifikation von grammatischer Information, die in den Wortformen kodiert ist (z.B. Kasus, Numerus, Tempus)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Elemente der morphologischen Struktur

- **Stämme**, Präfixe, Suffixe (in Flexion und Derivation)
  - Flexionsmorphologie:  
*frag+te+st*  
*ge+frag+t*
  - Derivationsmorphologie:  
*Er+kenn+ung*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Elemente der morphologischen Struktur

- Stämme, Präfixe, Suffixe (in Flexion und Derivation)
  - Flexionsmorphologie:  
*frag+te+st*  
*ge+frag+t*
  - Derivationsmorphologie:  
*Er+kenn+ung*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Elemente der morphologischen Struktur

- Stämme, Präfixe, Suffixe (in Flexion und Derivation)
  - Flexionsmorphologie:  
*frag+te+st*  
*ge+frag+t*
  - Derivationsmorphologie:  
*Er+kenn+ung*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Elemente der morphologischen Struktur

- **Grund-**, **Bestimmungswörter**, **Fugenelemente** (in der Komposition)  
*Sprach+erkennung+s+technik*  
*Sprach+erkennung+s+technik*
- Bestimmungswort: *Sprach* + Grundwort: *erkennung*
- Bestimmungswort: *Spracherkennung*  
+Fugenelement: *s* + Grundwort: *technik*
- Fugenelemente sind keine Flexionssuffixe. Sie sind aufgrund der phonologischen Struktur nur partiell vorhersagbar und müssen deshalb im Lexikon gelistet werden.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Elemente der morphologischen Struktur

- Stämme, Präfixe, Suffixe (in Flexion und Derivation)
- Grund-, Bestimmungswörter, Fugenelemente (in der Komposition)
- Modifikation von Stämmen (Umlaut, Ablaut)
  - *Mutter, Mütter*
  - *schwimmen, schwamm, geschwommen*
- Morpho-phonologische Regularitäten

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Morpho-phonologische Prozesse

- Systematische Modifikation, Einfügung und Tilgung von Lauten/Phonemen, die aus der phonetisch/phonologischen Struktur vorhersagbar ist:

Einfügung: *bad+st → badest*

Tilgung: *ras+st → rast*

- Ab- und Umlaut sind im Lexikon kodiert
- Morphophonologische Prozesse sind Angelegenheit der Grammatik (der morphologischen Analyse)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein Flexionsbeispiel aus dem Türkischen

- *Evlerinizdeyiz*
- *Ev+ler+iniz+de+yiz*
- *Haus+pl+poss-2.pers-pl+in+wir-sind*
- *"Wir sind in euren Häusern"*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Morphologische Verarbeitung in der Computerlinguistik

- Komponenten der Morphologischen Verarbeitung:
  - **Lemmatisierung:** Flexionsmorphologische Analyse: Ermittlung des **Stammes/Lemmas** („stemming“) und ggf. der in den Flexionsformen enthaltenen grammatischen Information
  - **Wortformengenerierung:** Erzeugung der (in Genus, Numerus, Kasus etc.) passenden Wortform zu einem Stamm
  - **Derivativ- und Komposita-Zerlegung:** Reduktion komplexer Wörter (Ableitungen und Zusammensetzungen) auf ihre Bestandteile
- Systeme zur morphologischen Verarbeitung sind für fast alle computerlinguistischen Anwendungen relevant (z.B. Rechtschreib- und Grammatikkorrektur; Informationszugriff; maschinelle Übersetzung).

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein Kompositionsbeispiel aus dem Deutschen

- Forstspezialrückeschlepper
- Forst+spezial+rücke+schlepper

Bei diesen Fahrzeugen handele es sich nämlich um Forstspezialrückeschlepper, deren Fahren als Beispiel in der Lohngr. W 7 Fallgr. 1 angeführt sei. (...) Ein Forstspezialschlepper, dessen Bestimmung das "Rücken" sei, sei nach den Regeln des allgemeinen Sprachgebrauchs ein Forstspezialrückeschlepper. Dabei spiele es auch keine Rolle, in welcher Reihenfolge die Bestandteile dieses Wortes verwendet seien. Mit Forstspezialrückeschlepper gleichbedeutend wäre auch "Spezialforstrückeschlepper", "Forstrückespezialschlepper" oder "Rückeforstspezialschlepper".

Aus einer Urteilsbegründung des Bundesarbeitsgerichtes

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Morphologische Verarbeitung in der Computerlinguistik

- Methodisches Werkzeug für alle Aufgaben morphologischer Verarbeitung sind „Endliche Automaten“.
- Wir betrachten die Verwendung endlicher Automaten an einer vergleichsweise einfachen Teilaufgabe der Lemma-Ermittlung.

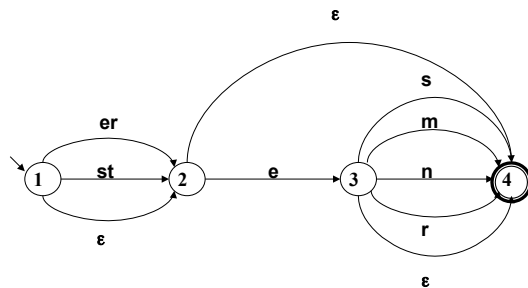
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Adjektivflexion: Paradigma (nur sog. „starke Flexion“)

klein+er	klein+e	klein+es	klein+e
klein+es/en	klein+er	klein+es/en	klein+er
klein+em	klein+er	klein+em	klein+en
klein+en	klein+e	klein+es	klein+e
klein+er+er	klein+er+e	klein+er+es	klein+er+e
klein+er+es/en	klein+er+er	klein+er+es/en	klein+er+er
klein+er+em	klein+er+er	klein+er+em	klein+er+en
klein+er+en	klein+er+e	klein+er+es	klein+er+e
klein+st+ er	klein+st+e	klein+st+ es	klein+st+e
klein+st+es/en	klein+st+er	klein+st+es/en	klein+st+er
klein+st+em	klein+st+er	klein+st+em	klein+st+en
klein+st+en	klein+st+e	klein+st+es	klein+st+e

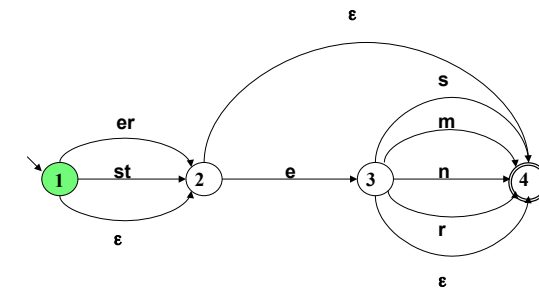
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Adjektivendungen: Kompakte Darstellung durch Zustandsdiagramm



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

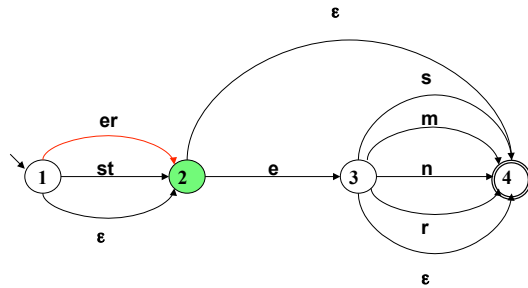
## Funktion des Zustandsdiagramms [1]



klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

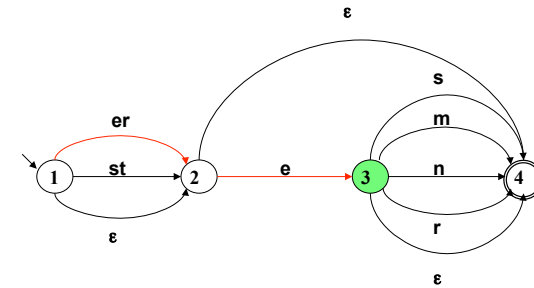
## Funktion des Zustandsdiagramms [2]



klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

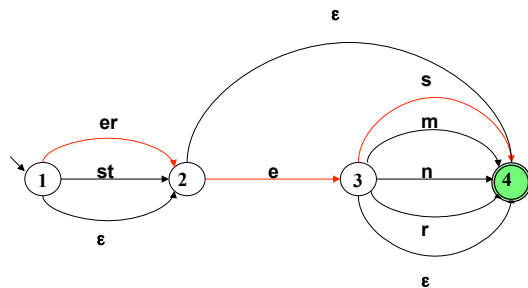
## Funktion des Zustandsdiagramms [3]



klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Funktion des Zustandsdiagramms [4]



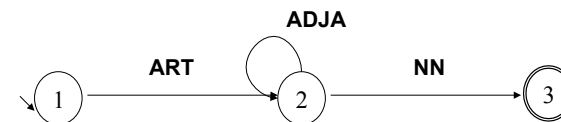
klein eres\_

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Zustandsdiagramme: Ein zweites Beispiel [1]

- Wortarten kombinieren sich in bestimmter Weise zu Satzteilen
- Um zu testen, ob eine Wortfolge in einem Dokument eine erlaubte Abfolge von Wortarten darstellt, können Zustandsdiagramme benutzt werden.
- Das folgende Zustandsdiagramm akzeptiert bestimmte einfache Nominalausdrücke, wie

*der Wagen*  
*eine interessante Vorlesung*  
*das neue schöne rote Dach*



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

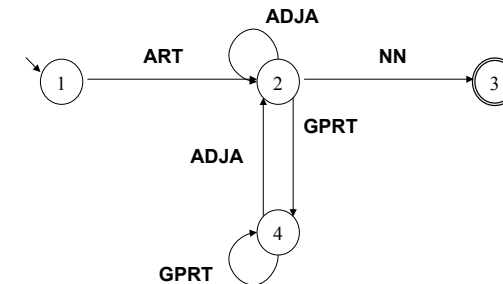
## Zustandsdiagramme: Ein zweites Beispiel [2]

- Das „Alphabet“ des Zustandsdiagramms sind Wortart-Bezeichnungen („Wortart-Tags“ oder auch „POS-Tags“, POS für „part of speech“, engl. für „Wortart“). Wir verwenden POS-Tags aus dem "Stuttgarter-Tübinger Tagset" (STTS), einer Standardnotation für das Deutsche:
  - ART (Artikel)
  - ADJA (adjektivisches Attribut)
  - NN (Gattungssubstantiv, Gattungsnomen)
- Erkannte „Worte“ sind erlaubte Abfolgen von Wortartensymbolen, z.B. „ART NN“, „ART ADJA NN“, „ART ADJA ADJA ADJA NN“
- Im Gegensatz zum Adjektivendungsdiagramm akzeptiert das Nominalausdrucksdiagramm beliebig lange Worte und beschreibt eine unendliche Sprache. Grund: Es enthält eine Schleife, es ist **zyklisch**.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Zustandsdiagramme: Ein zweites Beispiel [3]

Das abgebildete Diagramm akzeptiert auch Adjektive, die mit (einer oder mehreren) Gradpartikeln (GPRT) versehen sind, wie z.B. *eine ziemlich interessante Vorlesung*  
*das recht neue sehr sehr schöne rote Dach*



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Definitionen: Alphabet und Wort

- Ein **Alphabet**  $\Sigma$  ist eine endliche, nicht-leere Menge von Symbolen.
- Ein **Wort**  $w$  über dem Alphabet  $\Sigma$  ist eine endliche Kette von Symbolen aus  $\Sigma$ .
- Die **Wortlänge**  $|w|$  eines Wortes  $w$  ist die Anzahl der verketteten Symbole von  $w$ .
- Das **leere Wort**  $\epsilon$  ist das Wort mit Wortlänge 0 ( $|\epsilon|=0$ ).

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Definitionen: Sprache

- Ein **Sprache** über dem Alphabet  $\Sigma$  ist eine Menge von Worten über  $\Sigma$ .  
Zwei besondere Sprachen:
  - Die leere Wortmenge  $\emptyset$  heißt die „**leere Sprache**“.
  - Die maximale Sprache, die die Menge aller Worte über dem Alphabet  $\Sigma$  umfasst, ist  $\Sigma^*$  (der „**Stern**“ von  $\Sigma$ ).
- Anmerkung:  
Für jedes Alphabet  $\Sigma$  gilt:  $\epsilon \in \Sigma^*$ .

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik



## Beispiele

Beispiel 1:

$\Sigma = \{e, m, n, r, s, t\}$

$e, er, rrrrr, mnstmnst, \dots \in \Sigma^*$

$L = \{\varepsilon, e, er, em, en, es, ere, erer, erem, eren, eres, st, ste, stem, sten, ster, stes\}$

Beispiel 2:

$\Sigma = \{\text{ART}, \text{ADJA}, \text{NN}\}$

$L = \{\text{ART NN}, \text{ART ADJA NN}, \text{ART ADJA ADJA NN}\dots\}$

Alternative Formulierung:

$L = \{\text{ART ADJA}^n \text{NN} \mid n \in \mathbb{N}\}$

Bemerkungen:

- Mit  $\mathbb{N}$  bezeichnen wir hier die Menge der natürlichen Zahlen inklusive 0.
- $a^n$  ist die Kette, die durch n-faches Hintereinander-schreiben des Symbols  $a$  entsteht (für  $n=0$  ist  $a^n = \varepsilon$ )

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Beispiele

Beispiel 3:

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$L = \{x_1 \dots x_n y \mid n \in \mathbb{N}, x_i \in \Sigma \text{ für } 1 \leq i \leq n, y \in \{0, 5\}, n \in \mathbb{N}\}$

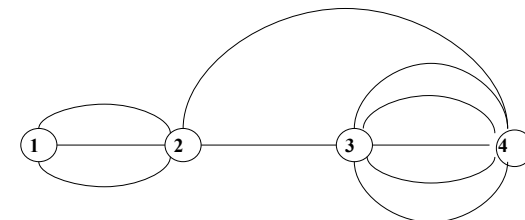
(die Menge der durch 5 teilbaren natürlichen Zahlen, wenn wir Ziffernfolgen mit 0-Präfixen ebenfalls zulassen)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Definition: Zustandsdiagramm [1]

Ein Zustandsdiagramm ist ein **gerichteter Graph mit Kantenbeschriftungen**.

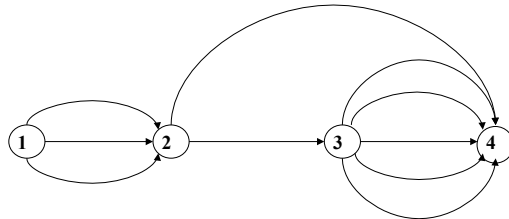
## Ein Graph



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

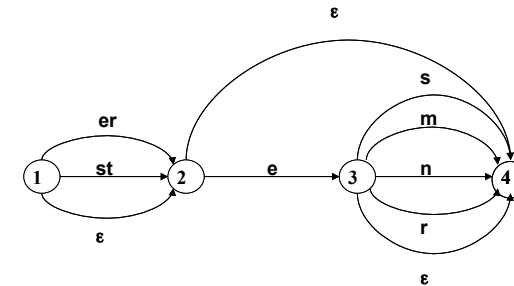
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein gerichteter Graph



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein gerichteter Graph mit Kantenbeschriftungen



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Zustandsdiagramm

Ein Zustandsdiagramm besteht aus

- **Knoten** (Zuständen) (im Beispiel: 1,2,3,4)
- davon ein **Startknoten** (1)
- einem oder mehreren **Endknoten** (4)
- **Kanten** zwischen den Knoten, die
  - **gerichtet** und
  - **beschriftet** sind
- Die Kanteninschriften bestehen aus Ketten von Symbolen über einem **Alphabet** (im Beispiel: e,r,m,n,s,t).
- Auch das **leere Wort** ( $\epsilon$ ) ist als Kantenbeschriftung zugelassen;  $\epsilon$  ist kein Symbol des Alphabets, sondern bezeichnet den Grenzfall der „aus 0 Symbolen bestehenden“ leeren Kette.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Definition: Zustandsdiagramm [2]

Formal wird ein Zustandsdiagramm definiert als ein Quintupel (Folge von 5 Elementen)

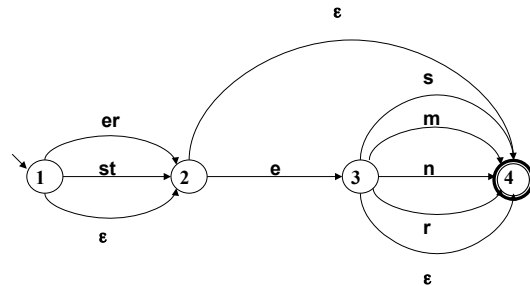
$A = \langle K, \Sigma, \Delta, s, F \rangle$ , wobei

- $K$  nicht-leere endliche Menge von Knoten (**Zuständen**)
- $\Sigma$  nicht-leeres **Alphabet**
- $s \in K$  **Startzustand**
- $F \subseteq K$  Menge von **Endzuständen**
- $\Delta : K \times \Sigma^* \times K$  Menge von beschrifteten Kanten (**Übergangsrelation**)

Anmerkung: Das Zustandsdiagramm heißt auch „**nicht-deterministischer endlicher Automat**“ (**NEA**), engl.: „non-deterministic finite-state automaton“ (**NFA**) (Erklärung später)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Adjektivendungen: Zustandsdiagramm



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Beispiel: Das Adjektivendungs-Diagramm

NEA  $A = \langle K, \Sigma, \Delta, s, F \rangle$  mit

- $K = \{1, 2, 3, 4\}$  (Zustände)
- $\Sigma = \{e, m, n, r, s, t\}$  (Alphabet)
- $s = 1$  (Startzustand)
- $F = \{4\}$  (einziger Endzustand)
- $\Delta = \{ \langle 1, er, 2 \rangle, \langle 1, st, 2 \rangle, \langle 1, \epsilon, 2 \rangle, \langle 2, e, 3 \rangle, \langle 2, \epsilon, 4 \rangle, \dots \}$  (Übergangsrelation)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Die Interpretation des Zustandsdiagramms

- Das Zustandsdiagramm beschreibt alle Symbolkombinationen oder „**Worte**“, die sich dadurch ergeben, dass man das Diagramm vom Startknoten zu einem Zielknoten durchläuft und die Inschriften der Kanten, die man dabei beschreitet, aufliest und aneinanderhängt. Man nennt die Menge der Worte, die sich so erzeugen lassen, die vom Diagramm beschriebene „**Sprache**“.
- Üblicherweise werden Diagramme verwendet, um Eingabeketten zu testen. Man spricht von **endlichen Automaten**, und sagt, dass ein Automat ein Wort **akzeptiert**.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Durch NEA akzeptiertes Wort/definierte Sprache

- Ein **Wort**  $w \in \Sigma^*$  wird durch den NEA  $A = \langle K, \Sigma, \Delta, s, F \rangle$  **akzeptiert** gdw. es eine Folge von Kanten (einen **Pfad** durch den NEA)  $\langle s, u_1, k_1 \rangle, \langle k_1, u_2, k_2 \rangle, \dots, \langle k_{n-1}, u_n, k_n \rangle$  gibt, sodass  $k_n \in F$  und  $u_1 u_2 \dots u_n = w$  (die **Konkatenation**, das Aneinanderhängen der Inschriften der durchlaufenen Kanten ergibt das Wort  $w$ ).
- Die vom NEA  $A = \langle K, \Sigma, \Delta, s, F \rangle$  **definierte** (akzeptierte) **Sprache**  $L(A)$  ist die Menge der von  $A$  akzeptierten Worte.

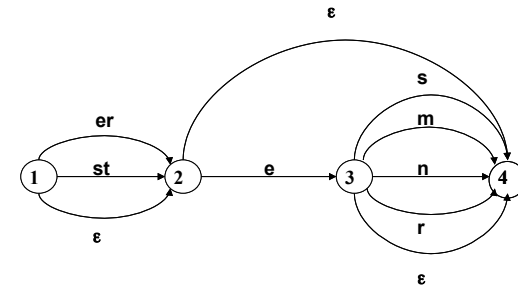
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

## Eine methodische Bemerkung

- Die Definition des Zustandsdiagramms/NEA spezifiziert eine **formale Notation**, die für sich genommen keine Bedeutung hat.
- Durch die Definitionen der letzten Folie (akzeptiertes Wort/definierte Sprache) wird diese Datenstruktur **interpretiert**: Wir verwenden Zustandsdiagramme, um die Zugehörigkeit von Symbolketten zu Sprachen zu definieren und zu testen.
- Hinzu kommen muss ein handhabbares **Verfahren**, ein **Algorithmus**, um den Zugehörigkeitstest tatsächlich durchzuführen.

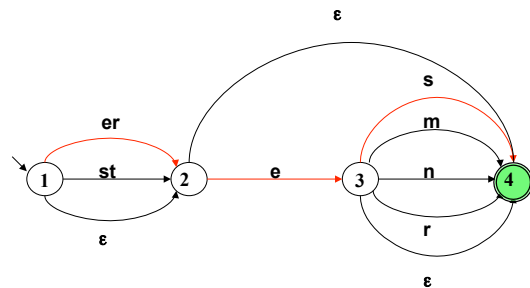
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein Suchproblem



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

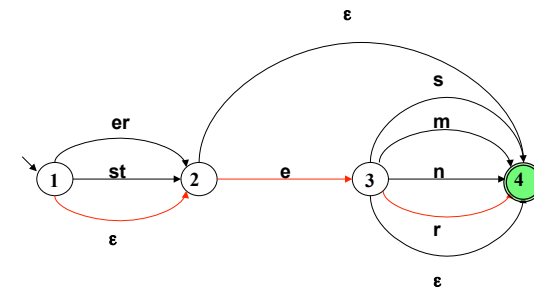
## Ein Weg durchs Diagramm



klein eres|

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein alternativer Weg durchs Diagramm



klein er|es

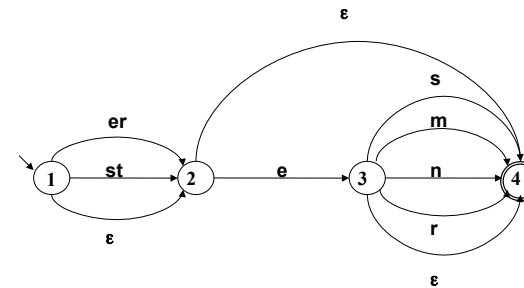
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Ein Suchproblem

- Das Diagramm erlaubt typischerweise an einem Knoten/ einem Zustand mehrere Übergänge bei derselben Eingabe (deshalb „nicht-deterministisch“).
- Die zufällige Wahl einer Kante kann sich erst viel später als falsch herausstellen. Sie gibt keine Gewähr, dass tatsächlich alle möglichen Wege durch das Diagramm getestet wurden.
- Wir benötigen ein Verfahren, das uns die vollständige Suche garantiert.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Pfadsuche: Startkonfiguration: Startzustand und Eingabewort auf der Agenda

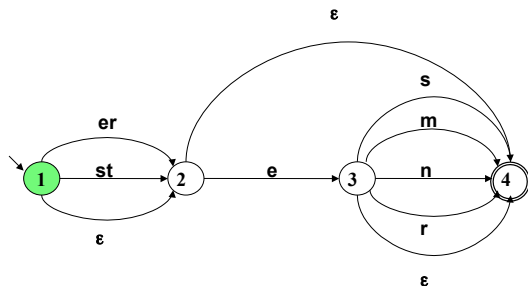


Eingabewort:

Agenda: 1 -- klein\_eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Pfadsuche: Nimm Aufgabe von der Agenda

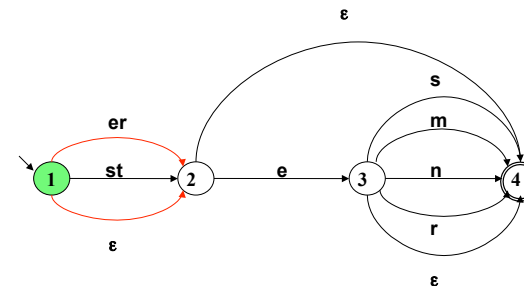


Eingabewort: klein\_eres

Agenda: \_\_\_\_\_

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

## Pfadsuche: Generiere neue Aufgaben

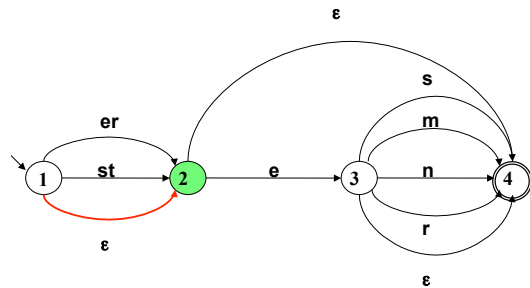


Eingabewort: klein\_eres

Agenda: 2 -- klein\_eres

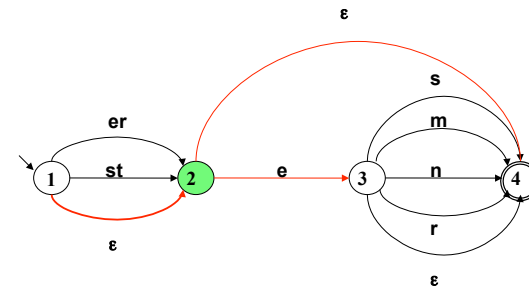
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

### Pfadsuche: Nimm Aufgabe von der Agenda



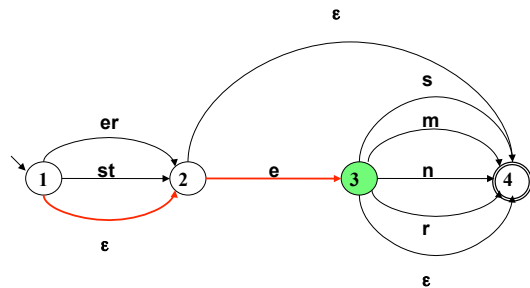
Eingabewort: klein eres      Agenda: 2 -- klein eres

### Pfadsuche: Generiere neue Aufgaben



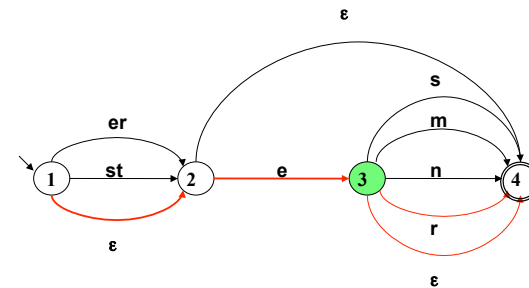
Eingabewort: klein eres      Agenda: 2 -- klein eres  
 3 -- klein eres  
 4 -- klein eres

### Pfadsuche: Nimm Aufgabe von der Agenda



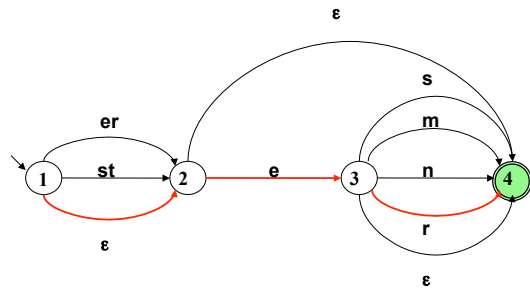
Eingabewort: klein eres      Agenda: 2 -- klein eres  
 4 -- klein eres

### Pfadsuche: Generiere neue Aufgaben



Eingabewort: klein eres      Agenda: 2 -- klein eres  
 4 -- klein eres  
 4 -- klein eres  
 4 -- klein eres

Pfadsuche: Nimm Aufgabe von der Agenda  
Keine neue Aufgabe!



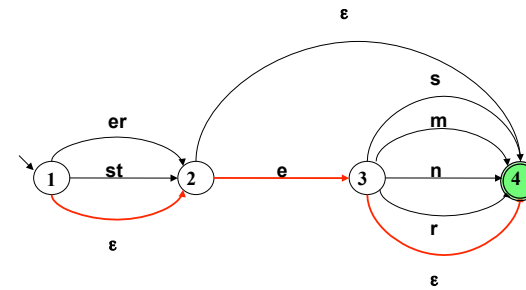
Eingabewort: klein eres

Agenda: 2 -- klein eres

4 -- klein eres  
4 -- klein eres  
2 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Pfadsuche: Nimm Aufgabe von der Agenda  
Keine neue Aufgabe!



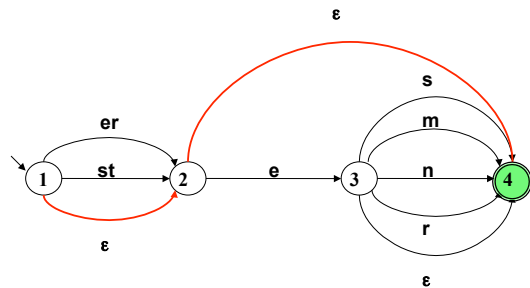
Eingabewort: klein eres

Agenda: 2 -- klein eres

4 -- klein eres  
2 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Pfadsuche: Nimm Aufgabe von der Agenda  
Keine neue Aufgabe!

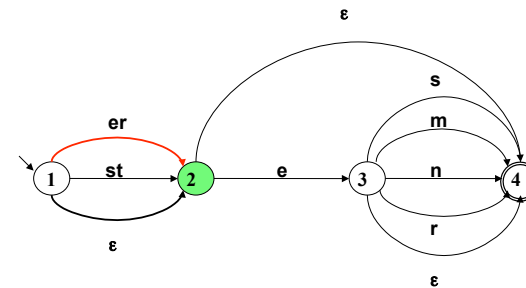


Eingabewort: klein eres

Agenda: 2 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Pfadsuche: Nimm Aufgabe von der Agenda  
Backtracking!

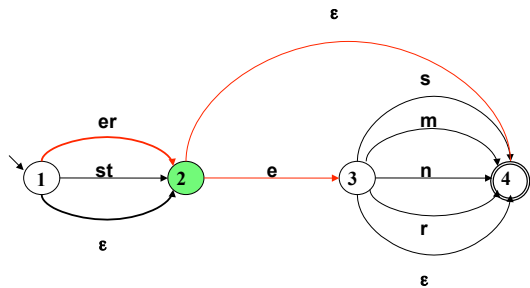


Eingabewort: klein eres

Agenda: \_\_\_\_\_

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

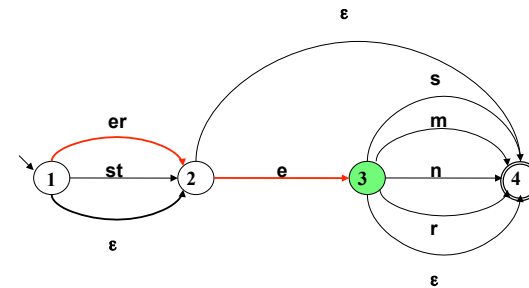
### Pfadsuche: Generiere Aufgaben



Eingabewort: klein eres      Agenda: 3 -- klein eres  
4 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

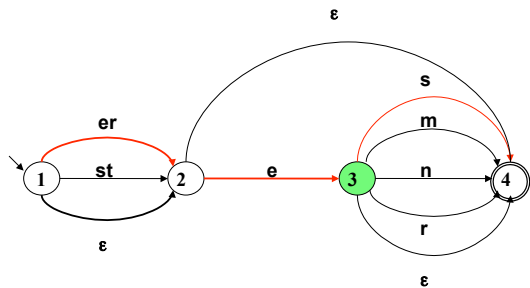
### Pfadsuche: Nimm Aufgabe von der Agenda



Eingabewort: klein eres      Agenda: 4 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

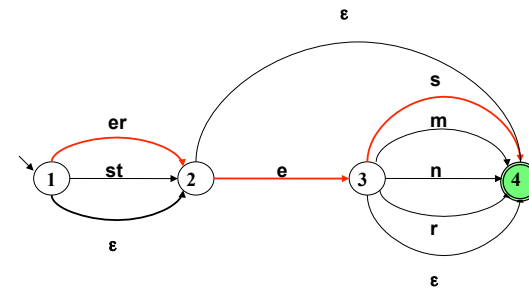
### Pfadsuche: Generiere Aufgabe



Eingabewort: klein eres      Agenda: 4 -- klein eres  
4 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

### Pfadsuche: Nimm Aufgabe von der Agenda: Eingabe abgearbeitet, Zielzustand: Akzeptiere!



Eingabewort: klein eres      Agenda: 4 -- klein eres

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik