



Frames

Sebastian Marius Kirsch

skirsch@moebius.inka.de





Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language





Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language



Zurück

Schliessen



Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language



Zurück

Schliessen



Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language





Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language



Zurück

Schliessen



Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language



Zurück

Schliessen



Überblick

- Einführung
- Grundlagen
- Frames im visuellen System
- Frame-Transformationen
- Frames und Sprache
- Frame Representation Language



Zurück

Schliessen



Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Einführung

- Wissensrepräsentationsformat
- Grundlage ist Gestaltpsychologie, Prototypentheorie der kognitiven Psychologie
- Marvin Minsky: Toshiba Professor of Media Arts and Sciences, MIT AI Lab, MIT Media Lab
- erstes Paper von 1974, sehr allgemein/bruchstückhaft
- Implementationen: KRL (Xerox, 1977), FRL (MIT, 1977) und andere





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können



Zurück

Schliessen



Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können





Grundlagen

- Datenstruktur, um stereotype Situationen zu repräsentieren
- hierarchisches Netzwerk
- höhere Level sind fest
- niedrigere Level haben *Terminale/Slots*, die nach Bedarf gefüllt werden
- Inhalt von Slots: *Sub-Frames*
- Slots enthalten Default-Belegungen
- Erwartungen und Beschränkungen an die Subframes, die einen Slot füllen können
- Suchprozess nach einem Frame, dessen Terminale vorhandene Subframes aufnehmen und zu einem größeren Ganzen zusammenbauen können



Zurück

Schliessen



Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





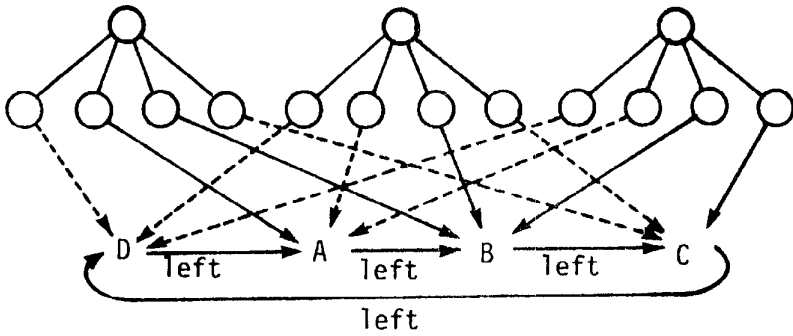
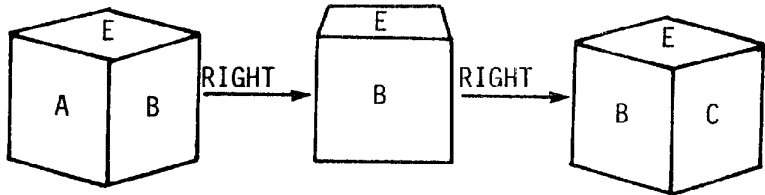
Frame-Systeme

- Mehrere Frames bilden ein *Frame-System*
- Frames eines Systems haben die gleichen Terminale
- erklärt, wie Information koordiniert wird
- über viele Probleme wird auf mehreren Ebenen nachgedacht
- Information muss nicht neu berechnet werden, sondern nur neuen Terminalen zugeordnet werden





Beispiel: Frames im visuellen System



Spatial Frames

⋮

Pictorial Frames

⋮

Relation Markers in common-terminal structure can represent more invariant (e.g. three-dimensional) properties.

- Flächen-Frame füllt Slots des Würfel-Frames
- Drehung des Würfels: Subframes werden neuen Slots zugeordnet

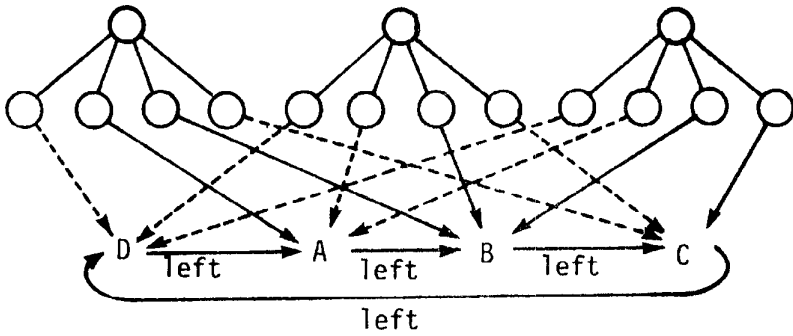
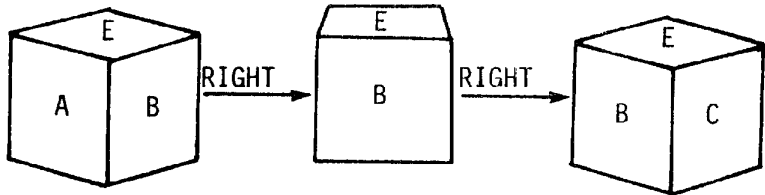


Zurück

Schliessen



Beispiel: Frames im visuellen System



Spatial Frames

⋮

Pictorial Frames

⋮

Relation Markers in common-terminal structure can represent more invariant (e.g. three-dimensional) properties.

- Flächen-Frame füllt Slots des Würfel-Frames
- Drehung des Würfels: Subframes werden neuen Slots zugeordnet

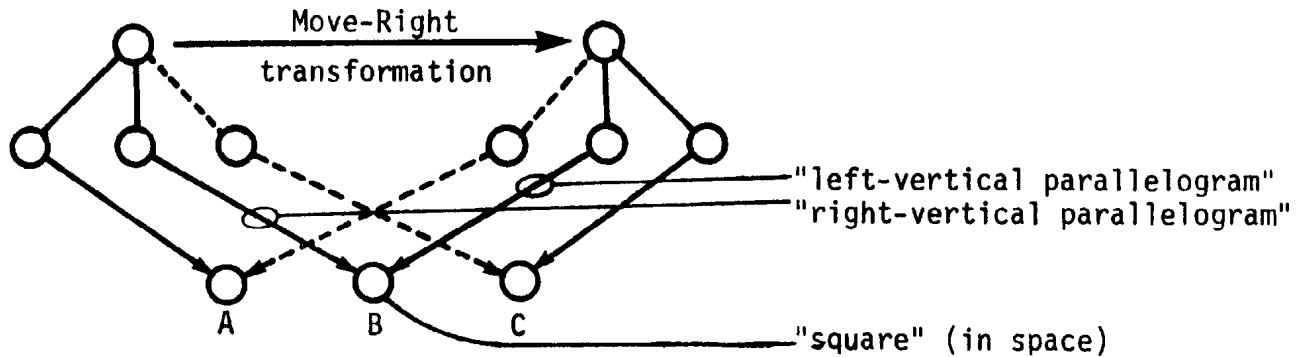


Zurück

Schliessen



Frame-Transformationen

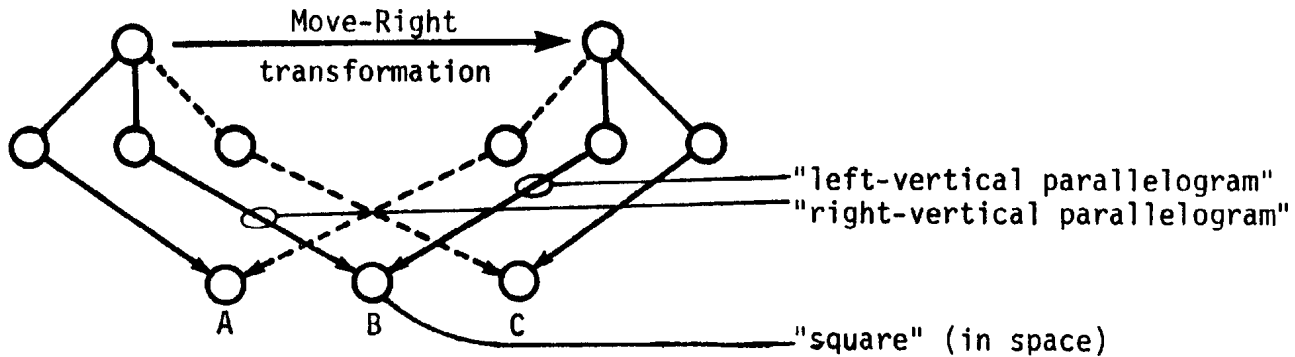


- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frame-Transformationen

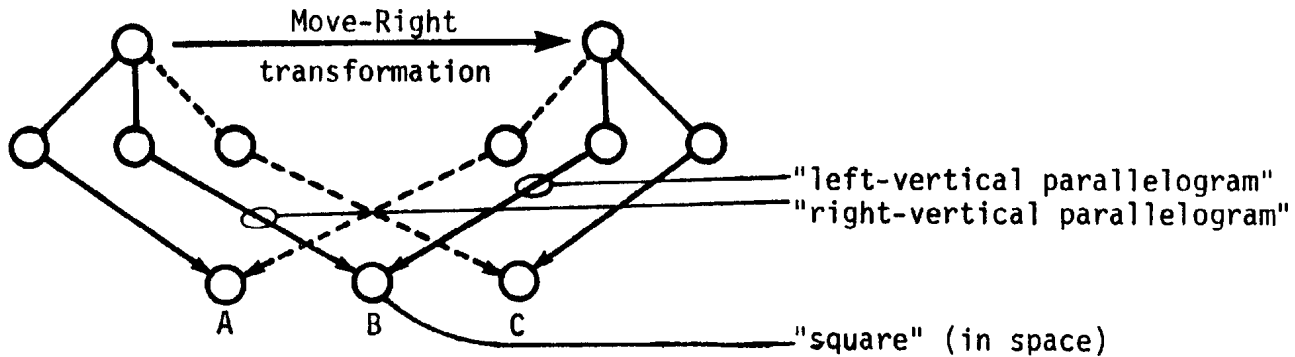


- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frame-Transformationen

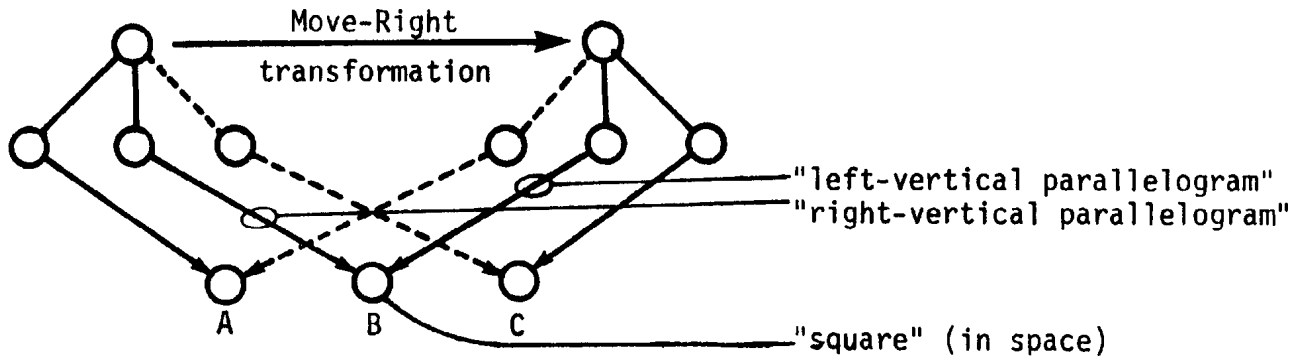


- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frame-Transformationen

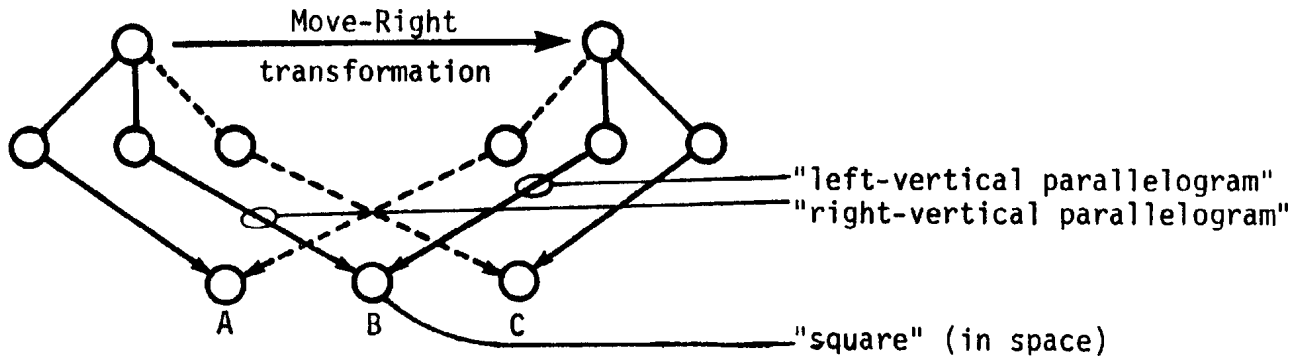


- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frame-Transformationen

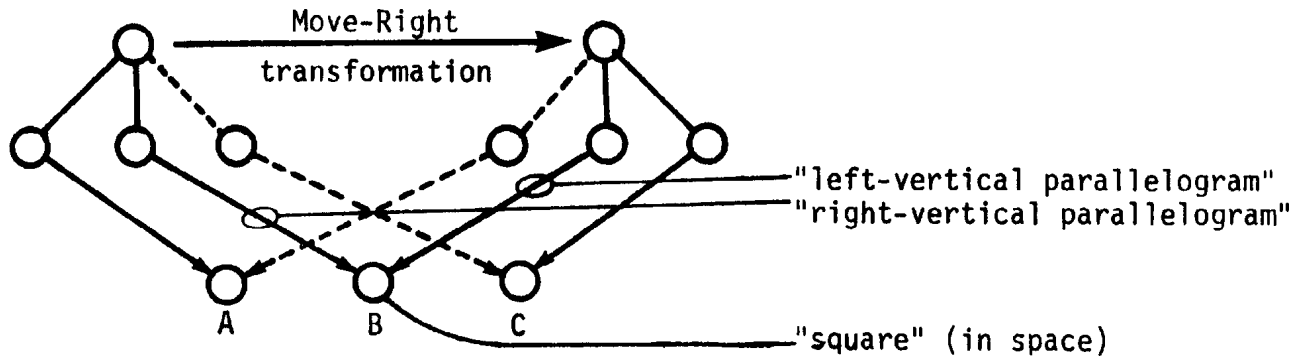


- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frame-Transformationen



- gelernte Regeln bauen Frames um
- Neuberechnung wird vermieden
- ähnlich zu Transformationsoperatoren in der Transformationsgrammatik
- logisches Denken und „gesunder Menschenverstand“ scheint ähnliches Vorgehen wie linguistische Transformationen zu benutzen
- ähnliche Mechanismen beim Denken wie bei der Kommunikation?





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache

- Frames, Constraints und Default-Belegungen können benutzt werden, um sprachliche Vorgänge zu beschreiben
- generative Grammatik entspricht der äusseren Erscheinung der sprachlichen Frames
- Operatoren der Transformationsgrammatik haben Ähnlichkeit mit Frame-Transformationen
- Idee: linguistische Aktivität umfasst größere Strukturen, als durch eine Grammatik beschrieben werden kann
- Syntax und Semantik sind keine Dichotomie, sondern zwei Extreme eines kontinuierlichen ganzen
- Nichtverstehen eines Satzes kann auf das Fehlschlagen eines semantischen Prozesses hindeuten





Frames und Sprache II

- bei nichtgrammatischen Sätzen findet kein Subframe genügend Fragmente, um gefüllt zu werden
- deshalb kann auch kein höherer Frame dem Satz eine Bedeutung zuordnen
- niedrige Level der syntaktischen Verarbeitung sind *verb-driven* (oder *head driven*)
- Bedeutung des Verbs tritt bei höheren Leveln in den Hintergrund





Frames und Sprache II

- bei nichtgrammatischen Sätzen findet kein Subframe genügend Fragmente, um gefüllt zu werden
- deshalb kann auch kein höherer Frame dem Satz eine Bedeutung zuordnen
- niedrige Level der syntaktischen Verarbeitung sind *verb-driven* (oder *head driven*)
- Bedeutung des Verbs tritt bei höheren Leveln in den Hintergrund





Frames und Sprache II

- bei nichtgrammatischen Sätzen findet kein Subframe genügend Fragmente, um gefüllt zu werden
- deshalb kann auch kein höherer Frame dem Satz eine Bedeutung zuordnen
- niedrige Level der syntaktischen Verarbeitung sind *verb-driven* (oder *head driven*)
- Bedeutung des Verbs tritt bei höheren Leveln in den Hintergrund





Frames und Sprache II

- bei nichtgrammatischen Sätzen findet kein Subframe genügend Fragmente, um gefüllt zu werden
- deshalb kann auch kein höherer Frame dem Satz eine Bedeutung zuordnen
- niedrige Level der syntaktischen Verarbeitung sind *verb-driven* (oder *head driven*)
- Bedeutung des Verbs tritt bei höheren Leveln in den Hintergrund





Frames und Sprache II

- bei nichtgrammatischen Sätzen findet kein Subframe genügend Fragmente, um gefüllt zu werden
- deshalb kann auch kein höherer Frame dem Satz eine Bedeutung zuordnen
- niedrige Level der syntaktischen Verarbeitung sind *verb-driven* (oder *head driven*)
- Bedeutung des Verbs tritt bei höheren Leveln in den Hintergrund





Frame Representation Language

- entwickelt 1977 von Roberts und Goldstein (MIT)
- spezialisierte Datenstruktur und LISP-Funktionen
- implementiert Defaults, Vererbung, Constraints, *procedural attachment* und Annotationen
- Quellcode (in MACLISP) erhältlich unter [6]





Frame Representation Language

- entwickelt 1977 von Roberts und Goldstein (MIT)
- spezialisierte Datenstruktur und LISP-Funktionen
- implementiert Defaults, Vererbung, Constraints, *procedural attachment* und Annotationen
- Quellcode (in MACLISP) erhältlich unter [6]





Frame Representation Language

- entwickelt 1977 von Roberts und Goldstein (MIT)
- spezialisierte Datenstruktur und LISP-Funktionen
- implementiert Defaults, Vererbung, Constraints, *procedural attachment* und Annotationen
- Quellcode (in MACLISP) erhältlich unter [6]





Frame Representation Language

- entwickelt 1977 von Roberts und Goldstein (MIT)
- spezialisierte Datenstruktur und LISP-Funktionen
- implementiert Defaults, Vererbung, Constraints, *procedural attachment* und Annotationen
- Quellcode (in MACLISP) erhältlich unter [6]





Frame Representation Language

- entwickelt 1977 von Roberts und Goldstein (MIT)
- spezialisierte Datenstruktur und LISP-Funktionen
- implementiert Defaults, Vererbung, Constraints, *procedural attachment* und Annotationen
- Quellcode (in MACLISP) erhältlich unter [6]





FRL Beispiele I: Frames erzeugen

```
(FASSERT MINSKY
```

```
  (NAME          ($VALUE ( |Marvin Minsky| )))
```

```
  (ADDRESS       ($VALUE ( |545 Technology Square, R...
```

```
  (PUBLICATIONS ($VALUE ( |A Framework for Represen...
```

```
  (INTERESTS     ($VALUE ( REPRESENTATION ))))
```

```
(FPUT 'MINSKY 'INTEREST '$VALUE 'ROBOTICS 'SOURCE: 'RBR)
```

```
(MINSKY
```

```
  ...
```

```
  (INTERESTS     ($VALUE ( REPRESENTATION )
```

```
    ( ROBOTICS (SOURCE: RBR))))))
```





FRL Beispiele II: Daten extrahieren

```
(FGET 'MINSKY 'INTERESTS '$VALUE)
```

```
-> ( REPRESENTATION ROBOTICS )
```

```
(FGET 'MINSKY 'INTERESTS '$VALUE 'ROBOTICS 'SOURCE:)
```

```
-> ( RBR )
```





FRL Beispiel III: Vererbung

```
(FASSERT MIT-AI
```

```
  (STREET ($VALUE ( |545 Technology Square| )))
  (CITY ($VALUE ( |Cambridge| )))
  (OFFICE ($DEFAULT ( 817 )))
  (ZIP ($VALUE ( |02139| ))))
```

```
(FASSERT MINSKY
```

```
  (NAME ($VALUE ( |Marvin Minsky| )))
  (AKO ($VALUE ( MIT-AI )))
  (OFFICE ($VALUE ( 821 )))
  (PUBLICATIONS ($VALUE ( |A Framework for Represen...
  (INTERESTS ($VALUE ( REPRESENTATION ))))
```

```
(FGET 'MINSKY 'CITY '$VALUE)
```

```
-> ( |Cambridge| )
```





Scripts and Plans

Sebastian Marius Kirsch
skirsch@moebius.inka.de





Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne





Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne



Zurück

Schliessen



Überblick

- Einführung
- Beispiel: Geschichten
- Geschichten ohne Script
- Das Restaurant-Script
- Pläne
- Elemente von Plänen
- Benannte Pläne





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Einführung

- Repräsentation von stereotypen Abläufen
- Roger Schank: John Evans Professor Emeritus der Informatik, Bildung und Psychologie, Northwestern University; Robert Abelson: Professur Emeritus der Psychologie, Yale University
- zwei Klassen von Wissen: spezialisiertes Wissen und allgemeines Wissen
- ein grosser Teil der alltäglichen Interaktion zwischen Menschen läuft nach gelernten Schemata, sog. *Scripts* ab
- Scripts sparen Verarbeitungszeit und schränken Erwartungen bei häufig erlebten Situationen ein
- angewendet in SAM (Script Applier Mechanism) von Cullingford, Lehnert, Gershman, Carbonell (1975): Programm liest Geschichten, generiert Zusammenfassungen in mehreren Sprachen und beantwortet Fragen zu den Geschichten





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten

- Scripts ermöglichen es, auch bei unvollständig erzählten Geschichten den kompletten Handlungsrahmen zu synthetisieren
- verläuft eine Geschichte entlang eines Scripts, werden fehlende Passagen kaum wahrgenommen.
- Beispiel:

John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

- fehlen wichtige Schlüsselszenen, die die Instantiierung eines Scripts hervorrufen, führt dies zu Verwirrung:

John went to a restaurant. He saw a waitress. He went home.





Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion



Zurück

Schliessen



Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion



Zurück

Schliessen



Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion





Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion





Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion





Beispiel: Geschichten ohne Script

- verweist eine Geschichte auf kein Script, so ist sie oft vollkommen unverständlich:

John was walking on the street. He thought of cabbages. He picked up a shoe horn.

- Geschichten müssen nicht auf ein Script verweisen, um verständlich zu sein:

John wanted a newspaper. He found one on the street. He read it.

- illustriert Unterschied zwischen logischem Denken und scriptgesteuerter sozialer Interaktion



Zurück

Schliessen



Das Restaurant-Script

(menu on table)/(W brings menu)
S **PTRANS** menu to S

(S asks for menu)
S **MTRANS** signal to W
W **PTRANS** W to table
S **MTRANS** 'need menu' to W
W **PTRANS** W to menu

W **PTRANS** W to table
W **ATRANS** menu to S

S **MTRANS** food list to CP(S)
* S **MBUILD** choice of F
S **MTRANS** signal to W
W **PTRANS** W to table
S **MTRANS** 'I want F' to W

W **PTRANS** W to C
W **MTRANS** (**ATRANS** F) to C

C **MTRANS** 'no F' to W
W **PTRANS** W to S
W **MTRANS** 'no F' to S
(go back to *) or
S **PTRANS** S to out of restaurant

C **DO** (prepare F script)
to next scene



Zurück

Schliessen



Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (BeispielRestaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Instantiierung von Scripts

- Scripts werden durch ihre Header instantiiert
- Verschiedene Typen von Headern:

Precondition Header (PH) Voraussetzungen, die für die Abarbeitung eines Scripts erfüllt sein müssen (Beispiel Restaurant: Akteur ist hungrig)

Instrumental Header (IH) aktuelles Skript ist nur „Mittel zum Zweck“, bietet den Hintergrund für eine andere Aktion

Locale Header (LH) Script wird durch eine bestimmte Örtlichkeit aktiviert (Beispiel Restaurant)

Internal Conceptualization Header (ICH) Ein beliebiges Konzept einer Geschichte kann ein neues Script instantiiieren.





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Pläne

- In Fällen, bei denen kein Script greift, wird ein allgemeinerer Mechanismus benutzt, der „Plan“.
- Verhältnis zwischen Plänen und Scripts ist nicht klar
- Pläne, die mehrfach ausgeführt wurden, können zu einem Script werden, müssen aber nicht
- genauso kann eine Situation, die nur ein einziges Mal erlebt wurde, zu einem Script internalisiert werden.
- Spezialfall: Benannte Pläne beschreiben allgemeine Strategien zur Problemlösung
- Primitive **MBUILD** in Scripts verweist auf das Erstellen eines Plans





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:

I-goals *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht

D-goals *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.

- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.

- Arten von **D-goals**:

- D-KNOW
- D-PROX
- D-CONT
- D-SOCCONT
- D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:

I-goals *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht

D-goals *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.

- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.

- Arten von **D-goals**:

- D-KNOW
- D-PROX
- D-CONT
- D-SOCCONT
- D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:

I-goals *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht

D-goals *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.

- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.

- Arten von **D-goals**:

- D-KNOW
- D-PROX
- D-CONT
- D-SOCCONT
- D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:

I-goals *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht

D-goals *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.

- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.

- Arten von **D-goals**:

- D-KNOW
- D-PROX
- D-CONT
- D-SOCCONT
- D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Elemente von Plänen

- Verschiedene Arten von Zielen:
 - I-goals** *instrumental goals*, einfache und stereotype Ziele, oft durch instrumentelle Scripts erreicht
 - D-goals** *delta goals*, Ziele, die Veränderungen beschreiben, die für das Erreichen des Hauptziels nötig sind.
- **D-goals** haben keinen eigenen Sinn, sondern dienen nur dem Erreichen der höheren Ziele.
- Arten von **D-goals**:
 - D-KNOW
 - D-PROX
 - D-CONT
 - D-SOCCONT
 - D-AGENCY





Benannte Pläne

- Schritt zwischen Script und Plan
- Beispiel:

$$\text{USE}(X) = \text{D} - \text{KNOW}(\text{LOC}(X)) \\ + \text{D} - \text{PROX}(X) + \text{I} - \text{PREP}(X) + \text{DO}$$

- allgemeine Problemlösungsstrategien werden so unter einem bestimmten Namen abrufbar gemacht
- Probleme mit rekursiven Plänen **ASK** → **FIND** → **ASK** → **FIND** → ...





Benannte Pläne

- Schritt zwischen Script und Plan
- Beispiel:

$$\text{USE}(X) = \text{D} - \text{KNOW}(\text{LOC}(X)) \\ + \text{D} - \text{PROX}(X) + \text{I} - \text{PREP}(X) + \text{DO}$$

- allgemeine Problemlösungsstrategien werden so unter einem bestimmten Namen abrufbar gemacht
- Probleme mit rekursiven Plänen **ASK** → **FIND** → **ASK** → **FIND** → ...





Benannte Pläne

- Schritt zwischen Script und Plan
- Beispiel:

$$\text{USE}(X) = \text{D} - \text{KNOW}(\text{LOC}(X)) \\ + \text{D} - \text{PROX}(X) + \text{I} - \text{PREP}(X) + \text{DO}$$

- allgemeine Problemlösungsstrategien werden so unter einem bestimmten Namen abrufbar gemacht
- Probleme mit rekursiven Plänen **ASK** → **FIND** → **ASK** → **FIND** → ...





Benannte Pläne

- Schritt zwischen Script und Plan
- Beispiel:

$$\text{USE}(X) = \text{D} - \text{KNOW}(\text{LOC}(X)) \\ + \text{D} - \text{PROX}(X) + \text{I} - \text{PREP}(X) + \text{DO}$$

- allgemeine Problemlösungsstrategien werden so unter einem bestimmten Namen abrufbar gemacht
- Probleme mit rekursiven Plänen $\text{ASK} \rightarrow \text{FIND} \rightarrow \text{ASK} \rightarrow \text{FIND} \rightarrow \dots$





Benannte Pläne

- Schritt zwischen Script und Plan
- Beispiel:

$$\text{USE}(X) = \text{D} - \text{KNOW}(\text{LOC}(X)) \\ + \text{D} - \text{PROX}(X) + \text{I} - \text{PREP}(X) + \text{DO}$$

- allgemeine Problemlösungsstrategien werden so unter einem bestimmten Namen abrufbar gemacht
- Probleme mit rekursiven Plänen **ASK** → **FIND** → **ASK** → **FIND** → ...





Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X MTRANS Q? tto Y

CP X BE(PROX(Y))

UP Y knows Q

MP Y wants to MTRANS Q to X

RES Y MTRANS Q to X



Zurück

Schliessen



Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X MTRANS Q? tto Y

CP X BE(PROX(Y))

UP Y knows Q

MP Y wants to MTRANS Q to X

RES Y MTRANS Q to X





Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(**PROX**(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X



Zurück

Schliessen



Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(PROX(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X



Zurück

Schliessen



Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(PROX(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X



Zurück

Schliessen



Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(PROX(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X





Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(**PROX**(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X





Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X MTRANS Q? tto Y

CP X BE(PROX(Y))

UP Y knows Q

MP Y wants to MTRANS Q to X

RES Y MTRANS Q to X





Planboxen

- Elemente

ACT Aktion

UP *uncontrollable precondition*

CP *controllable precondition*

MP *mediating precondition*

RES Resultat

- Planbox für **ASK**: löst **D-KNOW** auf

ACT X **MTRANS** Q? tto Y

CP X **BE**(**PROX**(Y))

UP Y knows Q

MP Y wants to **MTRANS** Q to X

RES Y **MTRANS** Q to X



Zurück

Schliessen



Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Fazit

- zwei Beispiele für frühe Formalismen zur Speicherung von menschlichem Wissen
- erfolgreiche Programme (vgl. SAM) auf Basis dieser Formalismen
- praktische Anwendbarkeit ist trotzdem beschränkt
- haben heute an Bedeutung verloren
- Verhältnis zwischen Formalismen und dem menschlichen Denken an sich ist ungeklärt





Literatur

- [1] Marvin A. Minsky: *A Framework for Representing Knowledge*. In: Patrick Henry Winston (ed.): *The Psychology of Computer Vision*. New York 1975. S. 211-277. <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-306.pdf>
- [2] Roger C. Schank, Robert P. Abelson: *Scripts, Plans, Goals and Understanding. An Inquiry into Human Knowledge Structures*. New York et al. 1977. Chapters 3-4.
- [3] Wolfgang Wahlster: *Frames*. In: Skript zur Vorlesung 'Einführung in die künstliche Intelligenz', Sommersemester 2002, Universität des Saarlandes, Saarbrücken. <http://w5.cs.uni-sb.de/ss02/materialien/KI-2002-131-138.pdf>
- [4] R. Bruce Roberts, Ira P. Goldstein: *The FRL Primer*. AI Memo 408, Juli 1977. <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-408.pdf>



- [5] R. Bruce Roberts, Ira P. Goldstein: The FRL Manual. AI Memo 409, Juli 1977. <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-409.pdf>
- [6] FRL: Frame Representation Language. <http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/kr/systems/frames/frl/frl.tgz>

