



Universidade do Minho

@ Paulo Cortez & Manuel Filipe Santos 2010

Aula 10: Dependencias Conceptuais e *Scripts* (guiões)

OBJECTIVOS:

- **Representar** conhecimento contido em frases da Linguagem Natural acerca de eventos



Dependências Conceptuais

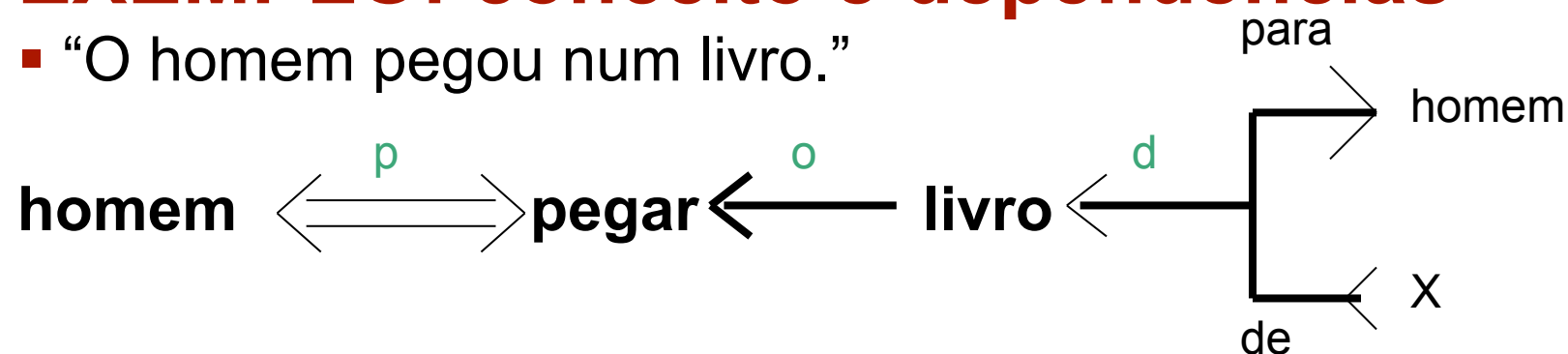
- Como lidar com a complexidade da **Linguagem Natural**?
- Os **grafos conceptuais** providenciam um conjunto de **primitivas** (com um determinado nível de detalhe) a partir das quais podemos construir peças de informação.
- Consegue-se assim representações com uma **semântica equivalente**, ou seja, várias frases de uma linguagem que tenham o mesmo significado têm uma mesma representação conceptual!
- Isto permite um **reconhecimento** (*parsing*) com uma **sintaxe** mínima!



Universidade do Minho

EXEMPLO: conceito e dependências

- “O homem pegou num livro.”



Significado dos símbolos:

 Ligação nos dois sentidos entre o ACTOR e a ACÇÃO

 Dependência e o seu sentido

passado tempo verbal passado p

objecto relação objecto o

direcção relação recipiente d



Universidade do Minho

Outro Exemplo:

“Eu dei um livro ao homem.”



Q1: Desenhar as dependências da frase.

(In-Class Teams 1 min)

- Grupos de 3 elementos, o que tiver os cabelos mais compridos é o representante (caneta e papel).



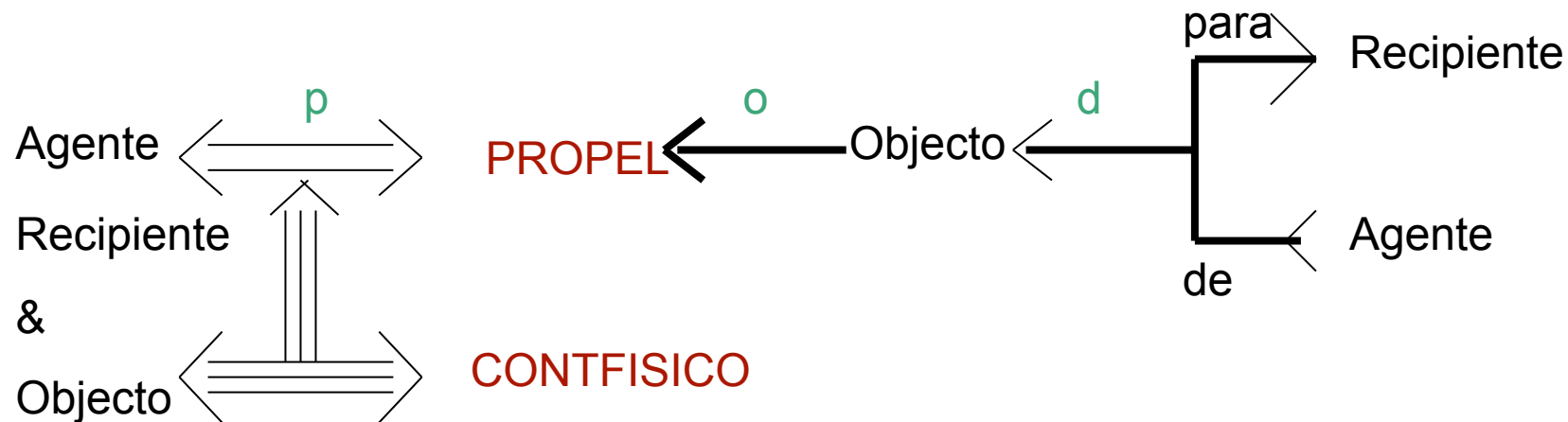
Exemplo de um conjunto de actos primitivos:

Dep. Concep.	Significado	Exemplos
atrans	transferência (posse)	dar, pegar, receber, vender, comprar
ptrans	transferência de lugar	ir, andar, mover, cair
mtrans	transferência mental	ler, dizer, esquecer, ensinar, prometer
ingest	colocar dentro	comer, beber, respirar
propel	aplicar força em	golpear, pontapear, bater
mbuild	construção mental	realizar, espantar, executar
grasp	elaborar actos	segurar, apanhar
move	mover parte do corpo	pontapear, empurrar
speak	elaborar verbos	dizer, contar, ...
attend	entradas sensoriais	ouvir, olhar
expel	enviar para fora	soprar, cuspir, espirrar, ...



Dicionário de Dependências Conceptuais

- Exemplo para o acto “golpear”:



- Esta entrada diz que o Agente do acto está a fazer uma PROPEL (dependência conceptual de aplicação de força)
- A seta extra (três linhas) indica que uma inferência (**exigência**) do verbo golpear como PROPEL é que o **Recipiente e Objecto têm de estar em contacto físico!**

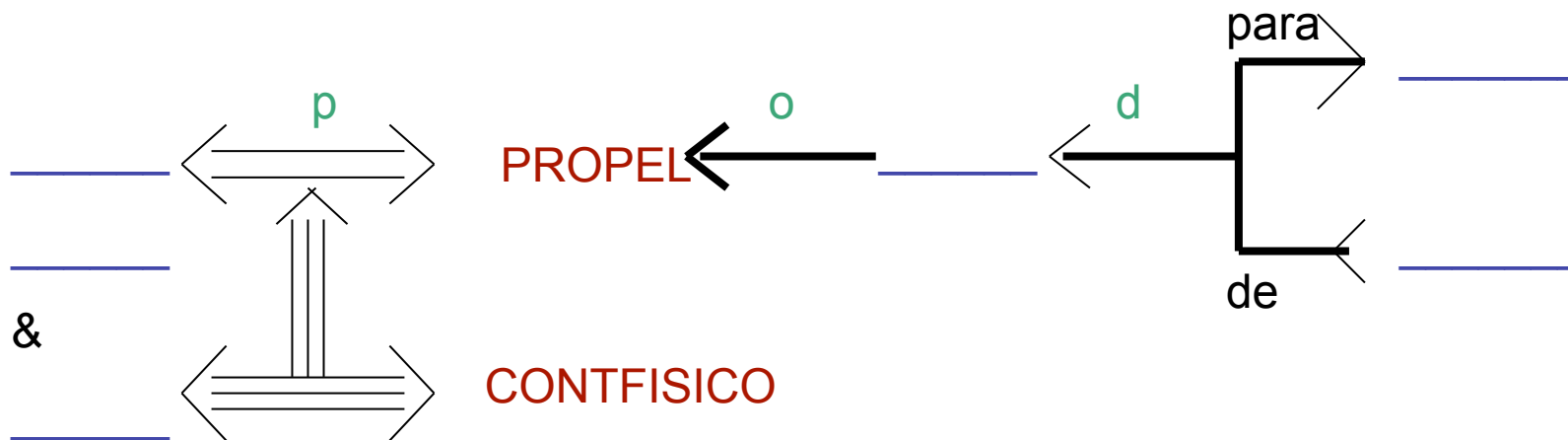


Universidade do Minho



Exemplo:

- “A Maria golpeou o João.”



Q2: Preencher os _____

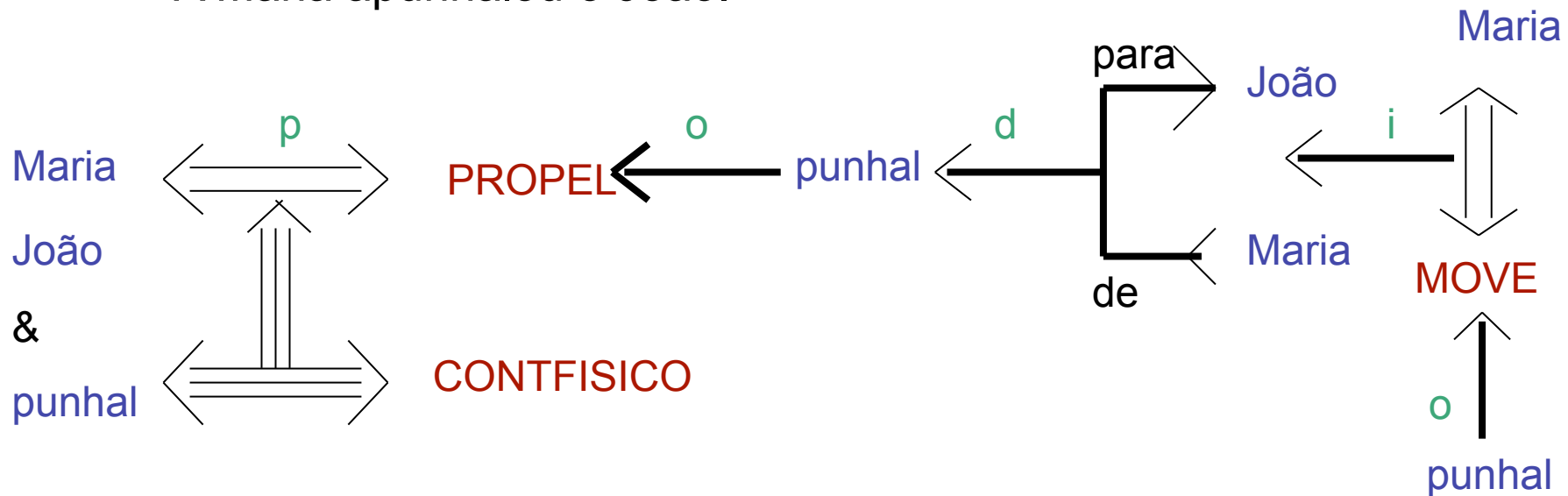
(In-Class Teams 1 min)

- Grupos de 3 elementos, o que tiver os cabelos menos compridos é o representante (caneta e papel).



Exemplo:

- “A Maria apunhalou o João.”



i corresponde ao INSTRUMENTO utilizado no acto

- O objectivo final é o reconhecimento (PARSING) semântico de frases escritas em Linguagem Natural. De seguida, podemos acrescentar inferências ao sistema. **Mas quais? E Quantas??**



SCRIPTS (guiões)

- O contexto indica quais as inferências razoáveis numa dada situação. Pode ser representado como uma estrutura de memória humana.

- Por exemplo:

“O José foi à Pizza Hut. E comeu uma marguerita. Depois foi ao cinema.”

Facilmente podemos fazer as inferências:

- Que tipo de lugar é a Pizza Hut?
- O que comeu o José?
- A quem o José pediu a comida?
- Que filme viu o José?
- ...

Fazemos isto porque possuímos conhecimento de **situações tipo (esterotipadas)**: organizamos o conhecimento, sabemos os comportamentos apropriados, descobrimos detalhes omitidos, ...



SCRIPTS (guiões)

- Mecanismo acerca de sequências de eventos;
- Estruturas que descrevem uma sequência estereotipada de eventos num contexto particular;
- Estruturas compostas por SLOTS (semelhantes aos FRAMES);

▪ Exemplo:

- Evento 1: O Actor vai a um restaurante
- Evento 2: O Actor vai da Porta até um Assento
- Evento 3: O Actor faz o Pedido ao Empregado
- Evento 4: O Actor come a Comida num Prato
- Evento 5: O Actor dá dinheiro ao Empregado
- Evento 6: O Actor deixa o Restaurante

- 1) Esta sequência de eventos genéricos é guarda na **memória**;
- 2) Quanto queremos **entender** a história específica, esta SCRIPT é **activada**;
- 3) Depois é **utilizada** para compreendermos a história.



Representação do Conhecimento - actos simples: uma Dependência Conceptual (DC)

- Em lógica de predicados, irá utilizar-se a seguinte notação (acção):

act(N, DC, [Agente, Objecto, De, Para]). % para cada evento!
N – número/id. do evento, DC – Dependência Conceptual

- Exemplo: “O João deu o livro à Maria.” é representado por:
act(1,atrans,[joao,livro,joao,maria]).



Dep. Concep.	Significado	Exemplos
atrans	transferência (posse)	dar, pegar, receber, vender, comprar
ptans	transferência de lugar	ir, andar, mover, cair
mtrans	transferência mental	ler, dizer, esquecer, ensinar, prometer
ingest	colocar dentro	comer, beber, respirar
propel	aplicar força em	golpear, pontapear, bater
mbuild	construção mental	realizar, espantar, executar
grasp	elaborar actos	segurar, apanhar
move	mover parte do corpo	pontapear, empurrar
speak	elaborar verbos	dizer, contar, ...
attend	entradas sensoriais	ouvir, olhar
expel	enviar para fora	soprar, cuspir, espirrar, ...

Q3: Represente: “O João foi ao cinema.”

(**In-Class Teams** 1 min) Grupos de 3 elementos, o que se sentou na sala em último lugar (do grupo) é o representante (caneta e papel).





Bases de Dados (situação actual) - Histórias : várias DC

- Em lógica de predicados, irá utilizar-se a seguinte notação:

story(N, [evento1, evento2, ...]).

N – identificação da história, DC – Dependência Conceptual

- Exemplo: **A história** “O João foi à Pizza Hut, comeu uma pizza e saíu.”

```
story(test1, [ act(A,ptrans,[joao,joao,_,pizza-hut]),  
               act(B,ingest,[_ ,pizza,_,_]),  
               act(C,ptrans,[_ ,_,_,_])  
             ]).
```

A, B e C são variáveis porque ainda não sei qual o id. (depende do script)!

- O objectivo do **PARSER** é detectar os actos, classificá-los segundo uma **DC primitiva** e **preencher os SLOTS** apropriados.



Representar Histórias – Exercício

Excerto de um diário de uma série de ficção científica:

“Zap foi ao Unirest, o restaurante que fica no fim do universo. Após ter comido um bife, deu uma gorgeta ao empregado e tele-transportou-se para a estação Beta.”

```
story(zap, [ act(A,_____, [ _____,unirest]),  
             act(B,ingest, [ _____]),  
             act(C,_____,[zap,gorgeta,zap,empregado]),  
             act(D,ptrans,[zap,zap,_____,_____])  
           ]).
```

Q4: Preencha os ____ (In-Class Teams 2 min)

Grupos de 3 elementos, o que foi representante menos vezes (do grupo) é o representante (caneta e papel).



Representação do Conhecimento – Base Conhecimento: SCRIPTS (situação tipo, o contexto)

- Em lógica de predicados, irá utilizar-se a seguinte notação:
script(Nome, [evento1, evento2, ...]).

Nome – id. da SCRIPT

- Exemplo: **Restaurante**

Evento 1: O Actor vai a um restaurante

Evento 2: O Actor vai da Porta até um Assento

Evento 3: O Actor faz o Pedido ao Empregado

Evento 4: O Actor come a Comida num Prato

Evento 5: O Actor dá dinheiro ao Empregado

Evento 6: O Actor deixa o Restaurante

```
script(restaurante, [ act(1,ptrans,[Actor,Actor,LugarX,Restaurante]),
                    act(2,ptrans,[Actor,Actor,Porta,Assento]),
                    act(3,mtrans,[Actor,Pedido,Actor,Empregado]),
                    act(4,ingest,[Actor,Comida,Prato,Actor]),
                    act(5,atrans,[Actor,Dinheiro,Actor,Empregado]),
                    act(6,ptrans,[Actor,Actor,Restaurante,LugarY])
                    ]).
```



Como encontrar uma SCRIPT?

- Relacionando cada SCRIPT na memória a uma palavra ou mais palavras:

trigger(Palavra,Script).

- Exemplo: **Restaurante**

trigger(pizza_hut,restaurante).
trigger(empregado,restaurante).



Sistema de Inferência em Prolog (dc.pl):

% Historia - nome da historia na Base de Dados

% Interp - como a historia foi entendida

```
demo(Historia,Interp):- story(Historia,Actos),  
                        infere(Actos,Interp).
```

```
infere(Actos,Interp):- encontra(Actos,NomeScript),  
                           script(NomeScript,Interp),  
                           unifica(Actos,Interp).
```

```
encontra(Historia,NomeScript)      % encontra uma Script cujo nome esteja  
:- member(act(_,_,Slots),Historia), % contido numa slot de um acto  
   member(Palavra,Slots),  
   nonvar(Palavra),  
   trigger(Palavra,NomeScript).
```



Sistema de Inferência em Prolog (dc.pl):

```
% unifica os actos da história com os actos da script
unifica([],_).
unifica([Ln|Rhistoria],[Ln|Rscript]):-unifica(Rhistoria,Rscript).
unifica(Historia,[_|Rscript]):- unifica(Historia,Rscript).

% mostra os resultados da interpretação
mostra(Historia):- demo(Historia,Interp), escreve(Interp).

escreve([]).
escreve([X|R]):- write(X),nl,escreve(R).
```



Exemplo em Prolog (restaurante.pl):

“O João foi à Pizza Hut, comeu uma pizza e saíu.”

```
:-[dc]. % chamar o sistema de inferência
% BD, tradução de: O João foi à Pizza Hut, comeu uma pizza e saíu:
story(test1,[act(A,ptrans,[joao,joao,_,pizza-hut]),
             act(B,ingest,[_ ,pizza,_,_]),
             act(C,ptrans,[_ ,_,_,_])
            ]).
% BC, script geral de ir a um restaurante:
script(restaurante, [ act(1,ptrans,[Actor,Actor,LugarX,Restaurante]),
                     act(2,ptrans,[Actor,Actor,Porta,Assento]),
                     act(3,mtrans,[Actor,Pedido,Actor,Empregado]),
                     act(4,ingest,[Actor,Comida,Prato,Actor]),
                     act(5,atrans,[Actor,Dinheiro,Actor,Empregado]),
                     act(6,ptrans,[Actor,Actor,Restaurante,LugarY])
                    ]).
trigger(pizza-hut,restaurante). % ligação do story ao script
```