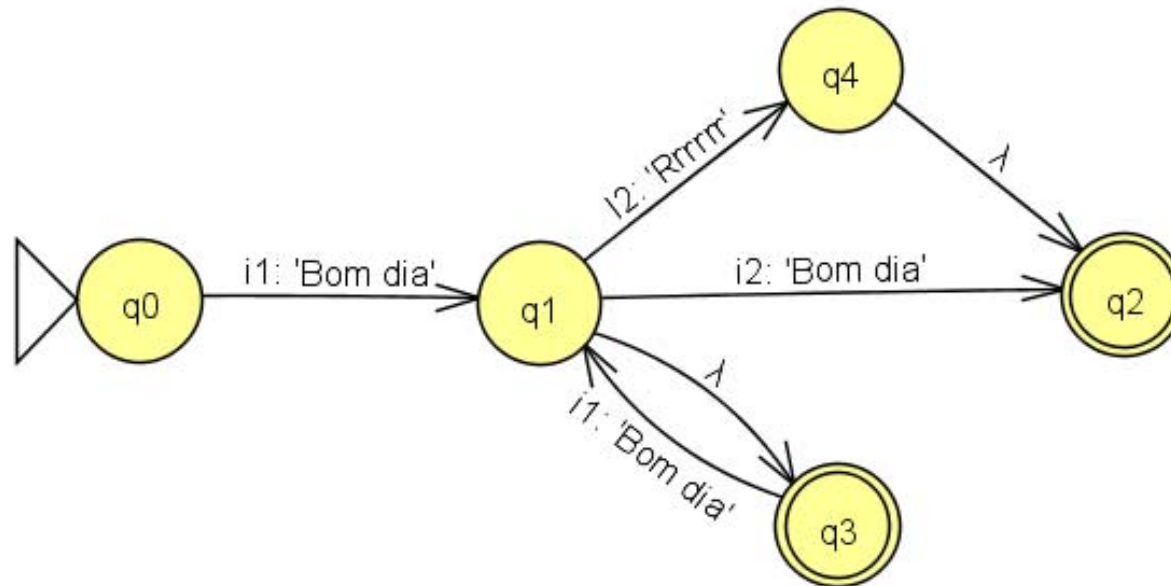


Linguística Computacional Interativa

Sintaxe, Semântica e Geração de Textos por Templates

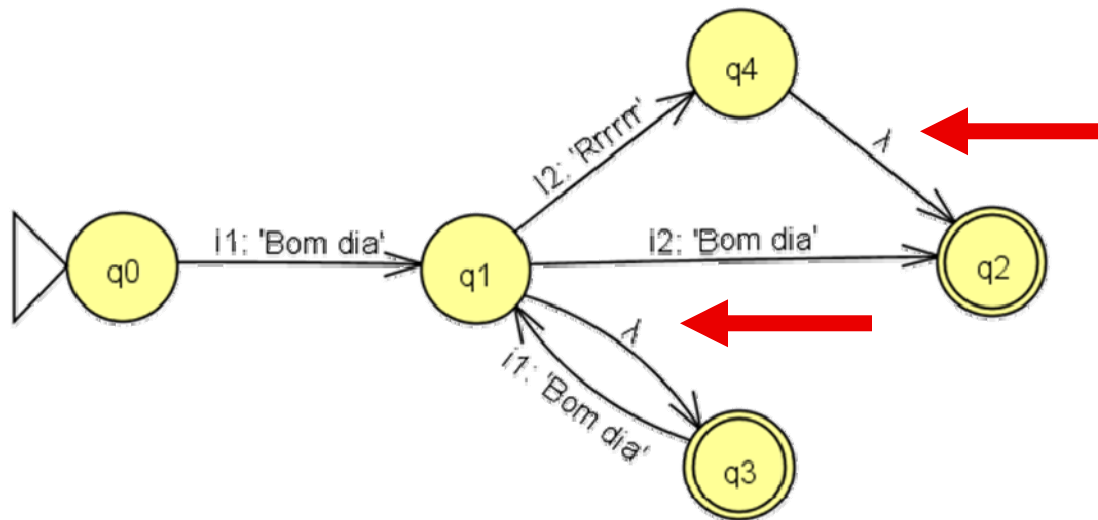
Aula de 18 de setembro de 2012

Especificação gramatical de modelo de 'conversa' ;-)



Sobre a escolha do autômato-modelo

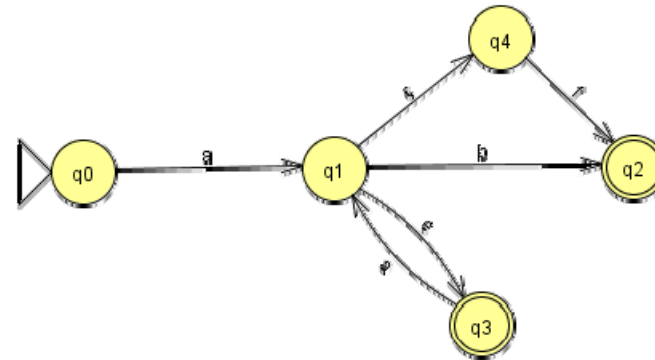
- Qual o efeito de λ ?



Máscaras de abstração / Reflexos Semânticos

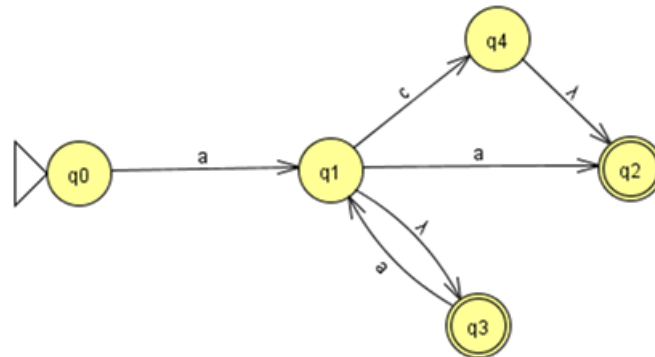
Abstração interlocutor+fala

- i1: 'Bom dia' $\rightarrow$ a
- i2: 'Bom dia' $\rightarrow$ b
- i2: 'Rrrr' $\rightarrow$ c
- i1: $\lambda \rightarrow$ e
- i2: $\lambda \rightarrow$ f



Abstração fala

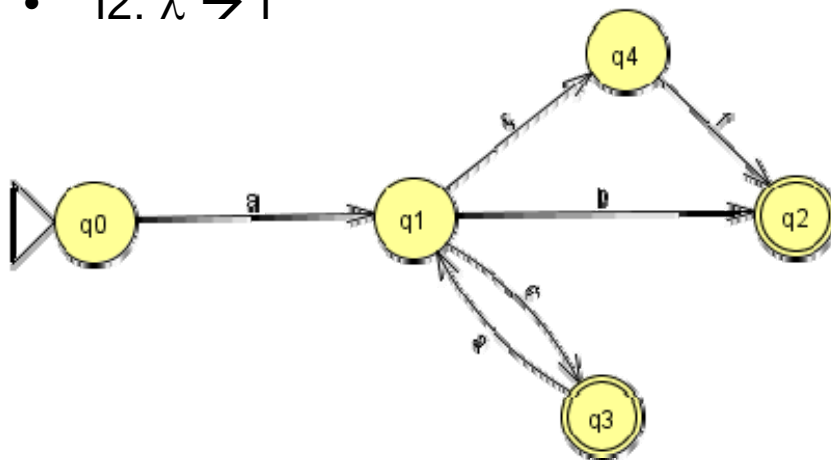
- i1: 'Bom dia' $\rightarrow$ a
- i2: 'Bom dia' $\rightarrow$ a
- i2: 'Rrrr' $\rightarrow$ b
- i1: $\lambda \rightarrow \lambda$
- i2: $\lambda \rightarrow \lambda$



Regras de Reescrita no Caso 1

Abstração interlocutor+fala

- i1: 'Bom dia' $\rightarrow$ a
- i2: 'Bom dia' $\rightarrow$ b
- i2: 'Rrrr' $\rightarrow$ c
- i1: $\lambda \rightarrow$ e
- i2: $\lambda \rightarrow$ f



- Gramática

$S \rightarrow aX$

$X \rightarrow cY$

$Y \rightarrow \lambda$

$X \rightarrow b$

$X \rightarrow \lambda$

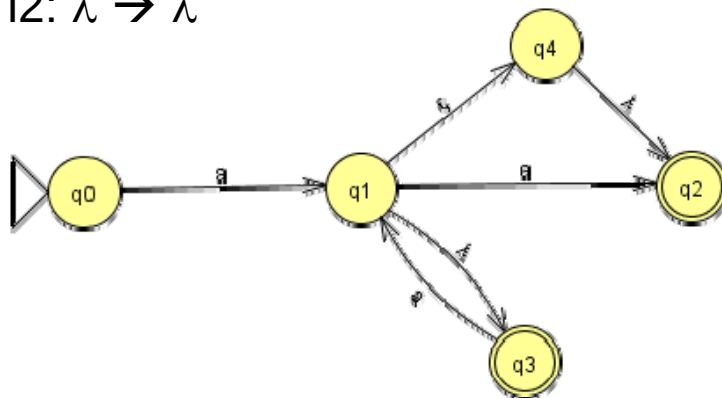
$X \rightarrow aX$



Regras de Reescrita no Caso 2

Abstração fala

- i1: 'Bom dia' $\rightarrow a$
- i2: 'Bom dia' $\rightarrow a$
- i2: 'Rrrr' $\rightarrow c$
- i1: $\lambda \rightarrow \lambda$
- i2: $\lambda \rightarrow \lambda$

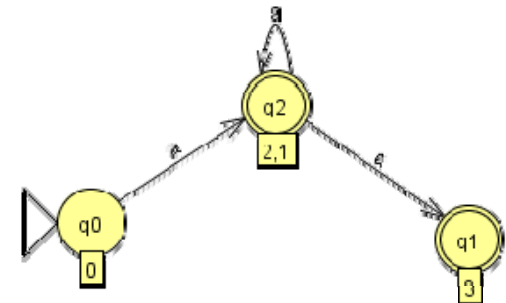


Autômato Mínimo e Gramáticas

$S \rightarrow Ac$
 $A \rightarrow Aa$
 $A \rightarrow a$

----- ou -----

$S \rightarrow aX$
 $X \rightarrow \lambda$
 $X \rightarrow aX$
 $X \rightarrow cY$
 $Y \rightarrow \lambda$



Pensando de novo: esta gramática está boa?

- CONVERSA → ‘Bom dia’, Bom dia’
- CONVERSA → ‘Bom dia’, REAÇÃO_AMBÍGUA
- CONVERSA → ‘Bom dia’, REAÇÃO_NEGATIVA
- REAÇÃO_AMBÍGUA → ‘ ‘, CONVERSA
- REAÇÃO_NEGATIVA → ‘Rrrr’, ‘ ‘

E esta? Melhor ou pior?

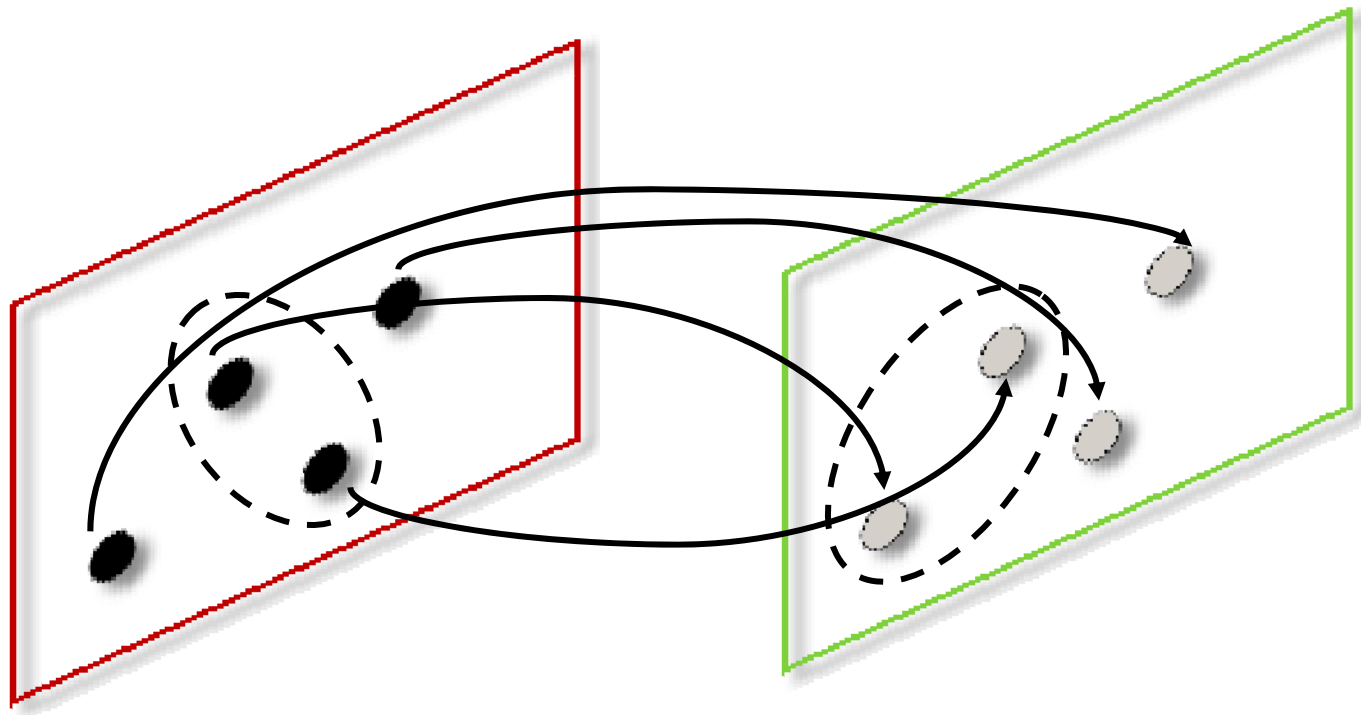
- CONVERSA → FALA_UM, FALA_OUTRO
- CONVERSA → FALA_UM, FALA_OUTRO, REAÇÃO
- FALA_UM → ‘Bom dia’
- FALA_OUTRO → ‘Bom dia’ | ‘Rrrr’ | ‘ ‘
- REAÇÃO → CONVERSA | ‘ ‘

E esta aqui? Como se compara às anteriores?

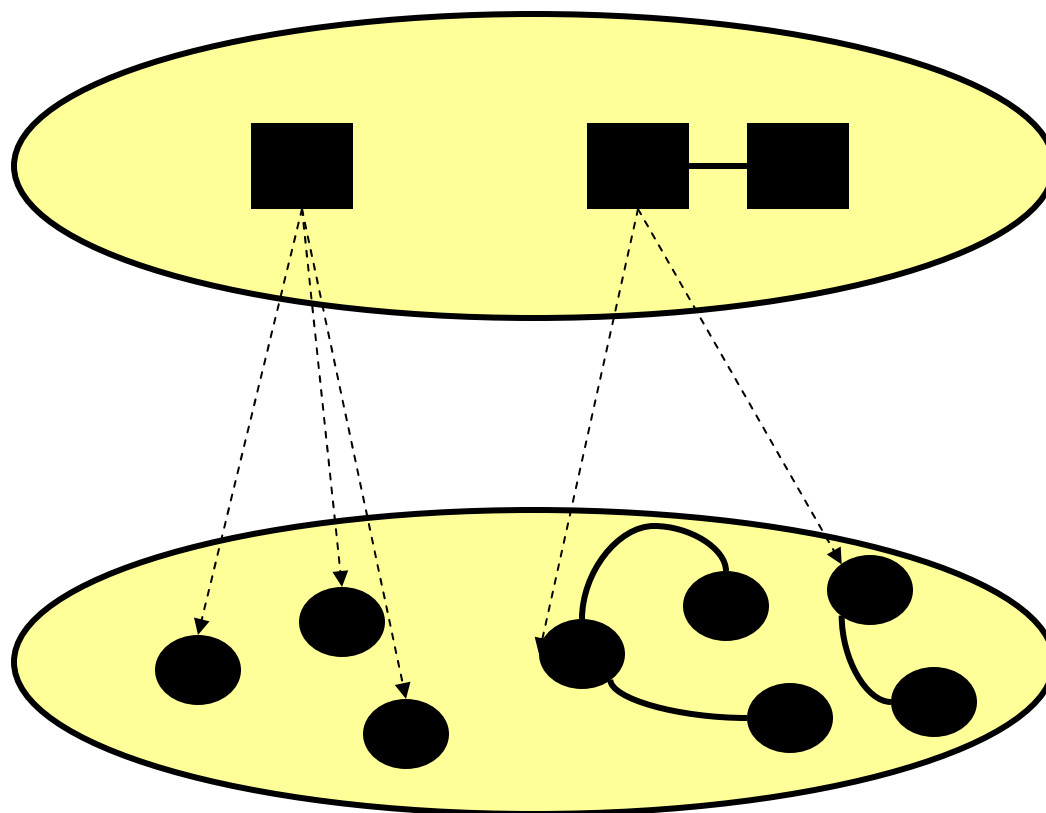
- CONVERSA → FALA_UM, FALA_OUTRO
- CONVERSA → FALA_UM, NÃO_FALA_OUTRO, CONVERSA
- CONVERSA → FALA_UM, ROSNA_OUTRO, CALA_UM
- FALA_UM → 'Bom dia'
- FALA_OUTRO → 'Bom dia'
- NÃO_FALA_OUTRO → ' '
- ROSNA_OUTRO → 'Rrrr'
- CALA_UM → ' '

“Gramáticas Semânticas”

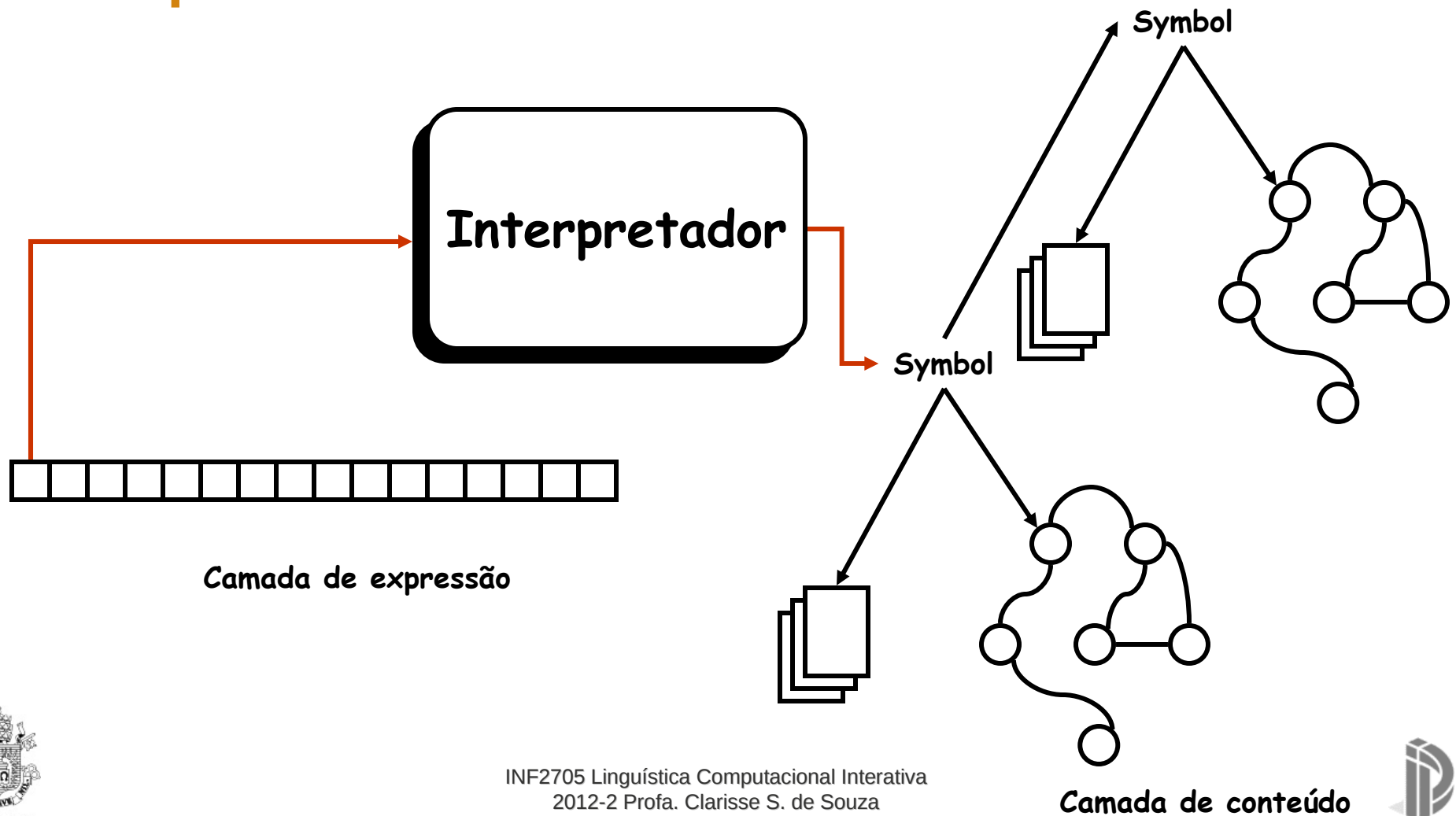
- Homomorfismos e isomorfismos no mapeamento entre estruturas sintáticas e representações semânticas



Sistemas de Tipos



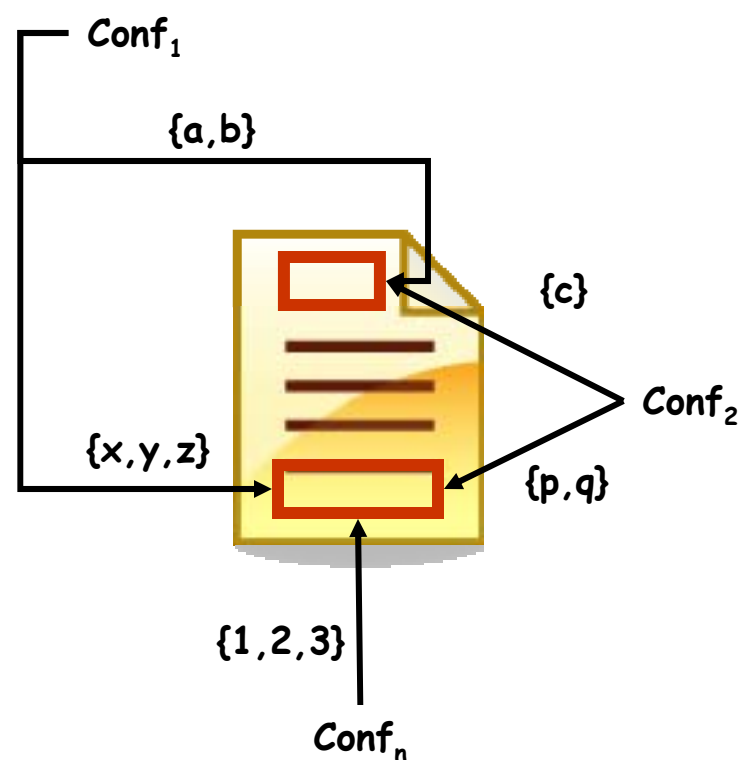
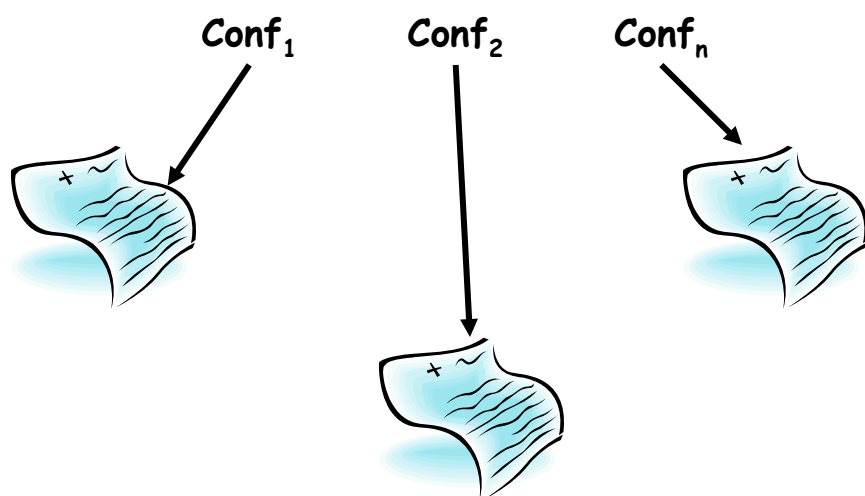
Interpretadores



Responda:

- Frequentemente os sistemas exibem para os usuários ‘texto enlatado’. Pensando nas relações entre dois sistemas de signos (um de superfície ou expressão e outro de significado ou conteúdo) vistas nos slides anteriores, o que podemos comentar acerca do *mapeamento* que se dá entre o texto enlatado (superfície ou expressão) e a configuração de sistema (significado ou conteúdo) associada à apresentação na interface?

Duas possibilidades



Níveis de processamento típicos em geração por *templates*

- Lexical
 - Escolha de vocábulos ou frases feitas (expressões)
- Morfo-sintático
 - Modulação vocabular de acordo com palavras vizinhas ou associadas

Exemplos de verbalizações de regras do AgentSheets

Conditions			loc			
see	SEE	1	0 0	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para si mesmo
		2	-1 0	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para cima
		3	-1 1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para cima à direita
		4	0 1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para a direita
		5	1 1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para baixo à direita
		6	1 0	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para baixo
		7	1 -1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para baixo à esquerda
		8	0 -1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para a esquerda
		9	-1 -1	<some-agent-name>	Se este agente vir	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO] ao olhar para cima à esquerda
see a	SEE-A	1	0 0	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente] ao olhar para si mesmo
		2	-1 0	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente] ao olhar para cima
		3	-1 1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		4	0 1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		5	1 1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		6	1 0	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		7	1 -1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		8	0 -1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
		9	-1 -1	<Some-agent-name>	Se este agente vir	qualquer das imagens do agente [nome-do-agente]
stacked	STACKED	1	immediately above	<some-agent-name>	Se este agente estiver imediatamente acima de	[INSERIR IMAGEM CORRESPONDENTE À DEPIÇÃO]

Enquanto o jogo está rodando, Lixo tem o seguinte comportamento:

- 1) Se este agente vir  ao olhar para a direita então ele apaga a si mesmo, olha para si mesmo e cria ali um novo agente com a aparência  e atribui a @valor o valor @valor + 1.
- 2) Se este agente vir  ao olhar para a esquerda então ele apaga a si mesmo, olha para si mesmo e cria ali um novo agente com a aparência  e atribui a @valor o valor @valor + 1.
- 3) Se este agente vir  ao olhar para cima então ele apaga a si mesmo, olha para si mesmo e cria ali um novo agente com a aparência  e atribui a @valor o valor @valor + 1.
- 4) Se este agente vir  ao olhar para baixo então ele apaga a si mesmo, olha para si mesmo e cria ali um novo agente com a aparência  e atribui a @valor o valor @valor + 1.

*WHILE-RUNNING ()

Insira um texto aqui para explicar o que este método faz!

Number of rules: 4

```

If
  SEE (  , 1 )
Then
  ERASE (  ), and
  NEW (  , 1 ), and
  SET ( @valor , para , @valor + 1 )

If
  SEE (  , 1 )
Then
  ERASE (  ), and
  NEW (  , 1 ), and
  SET ( @valor , para , @valor + 1 )

If
  SEE (  , 1 )
Then
  ERASE (  ), and
  NEW (  , 1 ), and
  SET ( @valor , para , @valor + 1 )

If
  SEE (  , 1 )
Then
  ERASE (  ), and
  NEW (  , 1 ), and
  SET ( @valor , para , @valor + 1 )

```