


# Semântica – parte I

*SCC5908 Tópicos em Processamento de Língua Natural*

Thiago A. S. Pardo



# Significado e representação

## Semântica

- **Significado** de palavras, orações, sentenças, textos
  - Atenção: há vários níveis de tratamento do significado
- Essencial para que sistemas de PLN sejam mais inteligentes
  - Exemplos?

3

## Semântica

- **Significado** de palavras, orações, sentenças, textos
  - Atenção: há vários níveis de tratamento do significado
- Essencial para que sistemas de PLN sejam mais inteligentes
  - De tarefas simples a complexas
    - Tradução e sumarização de textos
    - Geração e verificação de respostas de exames
    - Reação apropriada a ações, p.ex., acompanhar e desempenhar apropriadamente em um diálogo
    - Aprendizado automático
    - Perceber insultos, ironias, metáforas, etc.
    - Interpretar instruções

4

## Semântica

- **Análise semântica:** mapear superfície textual em significado
  - Dados linguísticos para não linguísticos
    - Expressões linguísticas para conceitos, proposições
- **Representação do significado**
  - Linguagens de representação do significado

5

## Semântica

- Exemplos de representação

*“Eu tenho um carro”*

- Lógica de 1ª ordem

- $\exists e, c \text{ Possuir}(e) \wedge \text{Possuidor}(e, \text{Falante}) \wedge \text{Possuído}(e, c) \wedge \text{Carro}(c)$

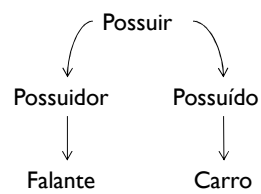
6

## Semântica

- Exemplos de representação

*“Eu tenho um carro”*

- Rede semântica



7

## Semântica

- Exemplos de representação

*“Eu tenho um carro”*

- Representação baseada em *frames*

- Possuir
  - Possuidor: Falante
  - Possuído: Carro

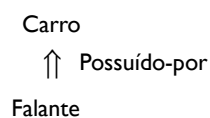
8

## Semântica

- Exemplos de representação

*“Eu tenho um carro”*

- Diagrama de dependência conceitual



9

## Semântica

- Linguagens de representação
  - Suposições **diferentes**
  - Perspectivas **variadas** da questão
  - Poder de representação **variado**
  - Fundamentos **em comum**
    - Símbolos que correspondem a objetos
    - Propriedades de objetos
    - Relações entre objetos

10

## Semântica

- Linguagens de representação
  - 2 aspectos
    - Representação do conteúdo linguístico
    - Representação do estado de coisas no mundo
  - Há requisitos desejáveis para as representações

11

## Requisitos da representação

- Verificabilidade
  - Deve ser possível verificar a veracidade de representações
    - Por exemplo, via constatação em uma base de conhecimento
  - Exemplo
    - Pergunta: *O restaurante serve comida vegetariana?*
    - Representação: `Serve(Restaurante, Comida_vegetariana)`
      - Se estiver na base, OK/VERDADE
      - Caso contrário, FALSO
        - Negativo ou não se sabe

12

## Requisitos da representação

- Evitar ambiguidade
  - Apesar de haver ambiguidades, a representação deve evitá-las
    - Nem sempre é possível
  - Exemplo
    - Sentença: *O homem viu a torre Eiffel enquanto voava.*
    - Representação:
      - $\text{Viu}(\text{Homem}, \text{Torre\_Eiffel}) \wedge \text{Momento}(\text{Viu}, \text{Enquanto\_voava})$
      - $\text{Viu}(\text{Homem}, \text{Torre\_Eiffel}) \wedge \text{Voava}(\text{Torre\_Eiffel})$

13

## Requisitos da representação

- Representação de vagueza
  - Interpretações abertas, mas não ambíguas
    - Exemplo
      - Sentença: *Eu quero comer comida italiana.*
      - O termo “comida italiana” é suficientemente específico para se decidir por um restaurante, por exemplo
      - ... mas é muito vago para saber o que se quer de fato comer

14

## Requisitos da representação

- **Forma canônica**
  - Mesmo significado por meio de diferentes expressões linguísticas, mas se deseja uma única representação
  - Exemplo
    - Várias sentenças, mesmo significado
      - O restaurante serve comida vegetariana?
      - Comida vegetariana é servida no restaurante?
      - O restaurante tem pratos vegetarianos?
      - Tem comida vegetariana no restaurante?
    - Idealmente, representação única
      - `Serve(Restaurante,Comida_vegetariana)`
    - Alternativamente, meio de se verificar compatibilidade entre representações
      - `Serve(Restaurante,Comida_vegetariana) = Tem(Restaurante,Pratos_vegetarianos)`

15

## Requisitos da representação

- **Forma canônica**
  - Mesmo significado por meio de diferentes expressões linguísticas, mas se deseja uma única representação
  - Exemplo
    - Várias sentenças, mesmo significado
      - O restaurante serve comida vegetariana?
      - Comida vegetariana é servida no restaurante?
      - O restaurante tem pratos vegetarianos?
      - Tem comida vegetariana no restaurante?
    - Idealmente, representação única
      - `Serve(Restaurante,Comida_vegetariana)`
    - Alternativamente, meio de se verificar compatibilidade entre representações
      - `Serve(Restaurante,Comida_vegetariana) = Tem(Restaurante,Pratos_vegetarianos)`

Por quê?

16



## Requisitos da representação

- Inferência e variáveis
  - Tirar conclusões sobre a veracidade de proposições que não são explicitamente representadas na base de conhecimento
  - Exemplos
    - Sentença: *Vegetarianos podem comer naquele restaurante?*
      - É preciso saber
        - “vegetarianos comem comida vegetariana”
        - “se aquele restaurante serve comida vegetariana”
    - Sentença: *Gostaria de encontrar um restaurante em que eu posso comer comida vegetariana.*
      - Não se cita nome de nenhum restaurante
      - Precisa-se de um elemento variável
        - `Serve(x, Comida_vegetariana)`

17

## Requisitos da representação

- Expressividade
  - Capacidade de se representar qualquer (ou uma grande variedade de) tipo de assunto/conhecimento
  - Qualquer “segmento textual” que faça sentido deve ser passível de representação
    - Restrição forte!

18

## Lógica de 1ª ordem

19

### Características

- Lógica de 1ª ordem
  - Flexível
  - Bem entendida
  - Computacionalmente tratável
  - Verificabilidade
  - Inferência
  - Expressividade
- *Cálculo de predicados*

20

## Elementos básicos

- **Termos:** representam objetos
  - Constantes (capitalizadas)
    - Restaurante, Comida\_vegetariana
  - Funções (podem indicar propriedades)
    - LocalDe(Restaurante)
  - Variáveis (não capitalizadas)
    - x, y, e, c
      - Necessitam de quantificadores

21

## Elementos básicos

- **Predicados:** representam relações entre objetos
  - Serve(Restaurante,Comida\_vegetariana)
  - Restaurante(Lanchonete\_da\_Maria)

22

## Elementos básicos

- Representações compostas, via conectivos lógicos
  - Eu tenho cinco reais e não tenho tempo.
  - $\text{Ter}(\text{Falante}, \text{Cinco\_reais}) \wedge \neg \text{Ter}(\text{Falante}, \text{Tempo})$

23

## Elementos básicos

- Conectivos lógicos
  - V = Verdade
  - F = Falso

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$
F	F	V	F	F	V
F	V	V	F	V	V
V	F	F	F	V	F
V	V	F	V	V	V

24

## Variáveis e quantificadores

- Variáveis
  - 2 possíveis usos
    - Objeto anônimo
    - Objetos de uma coleção
  - Uso de quantificadores
    - Existencial:  $\exists$  (lê-se “existe”)
    - Universal:  $\forall$  (lê-se “para todo”)

25

## Variáveis e quantificadores

- Quantificador existencial
  - Exemplo: objeto anônimo
    - *Um restaurante que serve comida mexicana perto do instituto.*
      - $\exists x \text{ Restaurante}(x) \wedge \text{Serve}(x, \text{Comida\_mexicana}) \wedge \text{PertoDe}(\text{LocalDe}(x), \text{LocalDe}(\text{Instituto}))$
      - Essa sentença será verdadeira se e somente se houver pelo menos um  $x$  que satisfaça todas as fórmulas (em uma base de conhecimento ou inferidas a partir da base)

26

## Variáveis e quantificadores

- **Quantificador universal**

- Exemplo: coleção de objetos

- *Todos os restaurantes vegetarianos servem comida vegetariana.*

- $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$

- Essa sentença só será verdadeira se toda substituição de  $x$  tornar a sentença verdadeira

27

## Variáveis e quantificadores

- **Quantificador universal**

- Exemplo: coleção de objetos

- $\text{RestauranteVegetariano}(\text{Natureba})_{\text{VERDADE}} \Rightarrow \text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})_{\text{VERDADE}}$   
???

- $\text{RestauranteVegetariano}(\text{MorraNatureza})_{\text{VERDADE}} \Rightarrow \text{Serve}(\text{MorraNatureza}, \text{Comida\_Vegetariana})_{\text{FALSO}}$   
???

- $\text{RestauranteVegetariano}(\text{Churrascada})_{\text{FALSO}} \Rightarrow \text{Serve}(\text{Churrascada}, \text{Comida\_Vegetariana})_{\text{FALSO}}$   
???

28

## Variáveis e quantificadores

- Quantificador universal

- Exemplo: coleção de objetos

- $\text{RestauranteVegetariano}(\text{Natureba})_{\text{VERDADE}} \Rightarrow$   
 $\text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})_{\text{VERDADE}}$   
 $\rightarrow \text{VERDADE}$

- $\text{RestauranteVegetariano}(\text{MorraNatureza})_{\text{VERDADE}} \Rightarrow$   
 $\text{Serve}(\text{MorraNatureza}, \text{Comida\_Vegetariana})_{\text{FALSO}}$   
 $\rightarrow \text{FALSO}$

- $\text{RestauranteVegetariano}(\text{Churrascada})_{\text{FALSO}} \Rightarrow$   
 $\text{Serve}(\text{Churrascada}, \text{Comida\_Vegetariana})_{\text{FALSO}}$   
 $\rightarrow \text{VERDADE}$

29

## Notação Lambda

- Útil para abstrair de uma fórmula em específico

- Uso do lambda ( $\lambda$ ) + variável + predicado

- Exemplo

- $\lambda x.P(x)$

30

## Notação Lambda

- Operação de “redução Lambda”
  - Instancia/especifica a fórmula com constantes
- Exemplo
  - $\lambda x.P(x)$  (constante A)  $\rightarrow P(A)$

31

## Notação Lambda

- Operação de “redução Lambda”
  - Instancia/especifica a fórmula com constantes
- Exemplo (2 etapas)
  - $\lambda x.\lambda y.PertoDe(x,y)$  (CasaDaMaria)  $\rightarrow$   
 $\lambda y.PertoDe(CasaDaMaria,y)$
  - $\lambda y.PertoDe(CasaDaMaria,y)$  (Centro)  $\rightarrow$   
 $PertoDe(CasaDaMaria, Centro)$

32



## Inferência

- Habilidade de
  - Adicionar novas proposições à base de conhecimento
  - Determinar a veracidade de proposições não explícitas na base de conhecimento
- Modus ponens
  - Um dos métodos mais usuais de inferência

33

## Inferência

- Modus ponens
  - $\alpha, \alpha \Rightarrow \beta \rightarrow \beta$ 
    - Ou seja, se  $\alpha$  é observado e se sabe que  $\alpha$  implica em  $\beta$ , então se pode deduzir  $\beta$

34

## Inferência

- Modus ponens

- Exemplo

- Base de conhecimento
  - RestauranteVegetariano(Natureba)
  - $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$
- Pode-se deduzir e adicionar na base
  - $\text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})$

35

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento progressivo (*forward chaining*)
  - Dos antecedentes para os consequentes das implicações
    - RestauranteVegetariano(Natureba)

36

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento progressivo (*forward chaining*)

- Dos antecedentes para os consequentes das implicações

- RestauranteVegetariano(Natureba) ⇒  
Serve(Natureba,Comida\_Vegetariana)

37

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento progressivo (*forward chaining*)

- **Vantagem:** muitas inferências podem ser feitas de antemão e aumentar a base de conhecimento, economizando tempo durante a consulta
- **Desvantagem:** pode gerar muitas proposições que nunca são necessárias

38

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento regressivo (*backward chaining*)

- Dos consequentes para os antecedentes das implicações
  - Primeiro se verifica se a consulta existe na base
  - Se não, buscam-se por implicações cujo lado direito case com a consulta
  - Tenta-se provar o lado esquerdo das implicações encontradas

39

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento regressivo (*backward chaining*)

- Base de conhecimento
  - RestauranteVegetariano(Natureba)
  - $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$
- Consulta
  - $\text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})$   
→ não está na base

40

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento regressivo (*backward chaining*)

- Base de conhecimento

- RestauranteVegetariano(Natureba)
- $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$

- Consulta

- $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$   
 $\rightarrow \text{RestauranteVegetariano}(\text{Natureba}) \Rightarrow$   
 $\text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})$

41

## Inferência

- Modus ponens

- Forma de raciocínio

- Encadeamento regressivo (*backward chaining*)

- Base de conhecimento

- RestauranteVegetariano(Natureba)
- $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$

- Consulta

- RestauranteVegetariano(Natureba)  $\Rightarrow$   
 $\text{Serve}(\text{Natureba}, \text{Comida\_Vegetariana})$   
 $\rightarrow$  antecedente está na base, então VERDADE

42

## Inferência

- **Cuidado**
  - Raciocínio com encadeamento regressivo vs. raciocínio regressivo
    - Raciocínio com encadeamento regressivo
      - Método confiável
    - Raciocínio regressivo (abdução)
      - Raciocínio plausível e útil muitas vezes, mas pode estar errado
        - Assume que, se consequente é verdade, antecedente é automaticamente verdade também

43

## Inferência

- **Cuidado**
  - Raciocínio regressivo (abdução)
    - Exemplo
      - Base de conhecimento
        - RestauranteVegetariano(Natureba)
        - $\forall x$  RestauranteVegetariano(x)  $\Rightarrow$  Serve(x,Comida\_Vegetariana)
      - Consulta
        - Serve(Churrascada,Comida\_Vegetariana)

44

## Inferência

- **Cuidado**
  - Raciocínio regressivo (abdução)
    - Exemplo
      - Base de conhecimento
        - RestauranteVegetariano(Natureba)
        - $\forall x \text{ RestauranteVegetariano}(x) \Rightarrow \text{Serve}(x, \text{Comida\_Vegetariana})$
      - Consulta
        - $\text{Serve}(\text{Churrascada}, \text{Comida\_Vegetariana})$
        - Pela implicação na base de conhecimento, o antecedente  $\text{RestauranteVegetariano}(\text{Churrascada})$  erroneamente assumido como verdadeiro

45

## Inferência

- **Cuidado**
  - Raciocínios com encadeamento progressivo e com encadeamento regressivo **não são completos**
    - Há inferências válidas que podem não ser encontradas por esses métodos de raciocínio
      - Há alternativas, mas mais caras computacionalmente
        - Evitadas, muitas vezes
        - Assumem-se os riscos dos raciocínios anteriores

46

## Exercício

- **Construa em lógica de 1ª ordem a representação de significado do trecho de texto abaixo**

*Ontem eu comprei um carro novo. É um Palio azul. Comprei em uma concessionária em São Carlos.*

47

## Exercício

- **Represente em lógica de 1ª ordem cada uma das sentenças abaixo**

*Eu comi.*

*Eu comi um sanduíche.*

*Eu comi um sanduíche em minha mesa.*

*Eu comi em minha mesa.*

*Eu comi um sanduíche de almoço.*

*Eu comi um sanduíche de almoço em minha mesa.*

48



## Exercício

- Possíveis respostas

*Comi(Eu)*

*Comi(Eu,Sanduíche)*

*Comi(Eu,Sanduíche,Minha\_mesa)*

*Comi(Eu,Minha\_Mesa)*

*Comi(Eu,Sanduíche,Almoço)*

*Comi(Eu,Sanduíche,Almoço,Minha\_mesa)*

49

## Problemas?

- Possíveis respostas

*Comi(Eu)*

*Comi(Eu,Sanduíche)*

*Comi(Eu,Sanduíche,Minha\_mesa)*

*Comi(Eu,Minha\_Mesa)*

*Comi(Eu,Sanduíche,Almoço)*

*Comi(Eu,Sanduíche,Almoço,Minha\_mesa)*

50

## Problemas

- Qual o número de argumentos do verbo “comer”?
  - Não há uma **aridade** fixa
- Como representar  **fatos sobre cada um dos argumentos**?
  - Por exemplo, onde fica a mesa, qual o tipo da comida, quais os tipos dos argumentos
- Como garantir que todas as **inferências válidas** podem ser derivadas diretamente da representação do evento?
  - Todos as representações se referem ao mesmo evento?
- Como garantir que todos os **eventos são do mesmo tipo**, mesmo que sejam eventos diferentes?
- Como garantir que **inferências inválidas** não sejam derivadas?
  - Eu comi a mesa? Ou na mesa?

51

## Possível solução

- **Predicados diferenciados**
  - Individualizam-se os predicados, resolvendo a questão do número diferente de argumentos

$Comi_1(Eu)$

$Comi_2(Eu, Sanduíche)$

$Comi_3(Eu, Sanduíche, Minha\_mesa)$

$Comi_4(Eu, Minha\_Mesa)$

$Comi_5(Eu, Sanduíche, Almoço)$

$Comi_6(Eu, Sanduíche, Almoço, Minha\_mesa)$

52

## Possível solução

- **Predicados diferenciados**

- **Mas nada indica a relação entre os eventos**
  - Por exemplo, se o evento numerado como 6 é verdade, todos os anteriores também deveriam ser

$Comi_1(Eu)$

$Comi_2(Eu, Sanduíche)$

$Comi_3(Eu, Sanduíche, Minha_mesa)$

$Comi_4(Eu, Minha_Mesa)$

$Comi_5(Eu, Sanduíche, Almoço)$

$Comi_6(Eu, Sanduíche, Almoço, Minha_mesa)$

53

## Possível solução

- **Postulados de significado**

- **Indicam como as coisas se relacionam, permitindo ligar os eventos**

Por exemplo:

$\forall w, x, y, z \text{ } Comi_6(w, x, y, z) \Rightarrow Comi_5(w, x, y)$

- **Mas há problemas de escalabilidade; só seria viável para domínios controlados/pequenos**

54

## Possível solução

- **Argumentos não especificados**

- Todos os argumentos são listados, mas não necessariamente especificados
  - Aridade fixa, conexão lógica entre eventos

$\exists w,x,y \text{ Comi}(Eu,w,x,y)$

$\exists x,y \text{ Comi}(Eu,\text{Sanduíche},x,y)$

$\exists x \text{ Comi}(Eu,\text{Sanduíche},x,\text{Minha\_mesa})$

$\exists w,x \text{ Comi}(Eu,w,x,\text{Minha\_Mesa})$

$\exists y \text{ Comi}(Eu,\text{Sanduíche},\text{Almoço},y)$

$\text{Comi}(Eu,\text{Sanduíche},\text{Almoço},\text{Minha\_mesa})$

55

## Possível solução

- **Argumentos não especificados**

- **Mas impõe restrições/compromissos muito fortes e fica difícil individualizar os eventos, caso seja interessante**

$\exists w,x,y \text{ Comi}(Eu,w,x,y)$

$\exists x,y \text{ Comi}(Eu,\text{Sanduíche},x,y)$

$\exists x \text{ Comi}(Eu,\text{Sanduíche},x,\text{Minha\_mesa})$

$\exists w,x \text{ Comi}(Eu,w,x,\text{Minha\_Mesa})$

$\exists y \text{ Comi}(Eu,\text{Sanduíche},\text{Almoço},y)$

$\text{Comi}(Eu,\text{Sanduíche},\text{Almoço},\text{Minha\_mesa})$

56

## Possível solução

- **Variáveis para eventos**

- Permitem referenciar eventos e individualizá-los, se necessário

$\exists e \text{ Comi}(e, \text{Eu}, \text{Sanduíche}, \text{Almoço}, \text{Minha\_mesa})$

57

## Possível solução

- **Variáveis para eventos**

- Permitem referenciar eventos e individualizá-los, se necessário

$\exists e \text{ Comi}(e, \text{Eu}, \text{Sanduíche}, \text{Almoço}, \text{Minha\_mesa})$

- Caso haja suposições adicionais, é possível incorporá-las

$\exists e \text{ Comi}(e, \text{Eu}, \text{Sanduíche}, \text{Almoço}, \text{Minha\_mesa}) \wedge \text{Momento}(e, \text{Ontem})$

58

## Possível solução

- Variáveis para eventos
  - Também oferecem uma alternativa para a questão do número variável de argumentos dos verbos

$$\exists e \text{ Comer}(e) \wedge \text{Quem\_comeu}(e, \text{Eu}) \wedge \text{O\_que\_comeu}(e, \text{Sanduíche}) \wedge \\ \text{Refeição}(e, \text{Almoço}) \wedge \\ \text{LocalDe}(e, \text{Minha\_mesa}) \wedge \text{Momento}(e, \text{Ontem})$$

59

## Tempo

- Representação na lógica de 1ª ordem
  - Lógica temporal
    - Como representar expressões de tempo
    - Conceitos importantes
      - Ponto no tempo
        - Incluindo ponto de início e ponto de término
      - Intervalo de tempo
      - Linha do tempo

60

## Tempo

- Representação na lógica de 1ª ordem
  - Lógica temporal

### Exemplos

*Eu cheguei em São Paulo.*

$\exists e \text{ Chegar}(e) \wedge \text{Quem}(e, \text{Eu}) \wedge \text{Onde}(e, \text{São\_Paulo}) \wedge \text{Precede}(e, \text{Agora})$

*Eu estou chegando em São Paulo.*

$\exists e \text{ Chegar}(e) \wedge \text{Quem}(e, \text{Eu}) \wedge \text{Onde}(e, \text{São\_Paulo}) \wedge \text{MembroDe}(e, \text{Agora})$

*Eu vou chegar em São Paulo.*

$\exists e \text{ Chegar}(e) \wedge \text{Quem}(e, \text{Eu}) \wedge \text{Onde}(e, \text{São\_Paulo}) \wedge \text{Precede}(\text{Agora}, e)$

61

## Tempo

- Representação na lógica de 1ª ordem
  - Lógica temporal

- É importante diferenciar

- Tempo do evento (E)
- Tempo da sentença (S)
- Tempo de referência (R)

- Exemplo: *Quando o vôo partiu, ela comeu.*

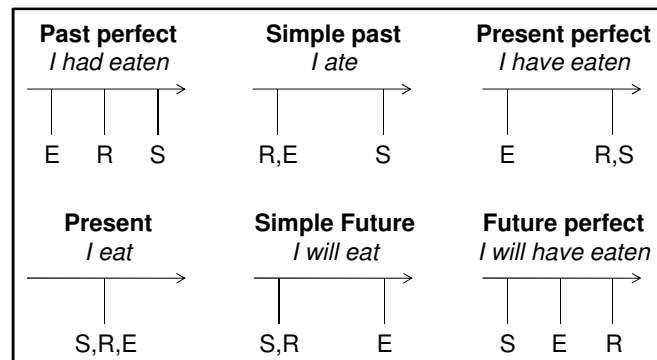
- E: passado
- S: após o tempo do evento
- R: partida do vôo

62

## Tempo

- Representação na lógica de 1ª ordem
  - Lógica temporal

Proposta de Reichenbach (1947) para o inglês: tempos do inglês & ordenação de tempos sobre linha do tempo



## Eventos

- Tipos de eventos
  - Definidos pelo momento da ocorrência, intervalo de tempo de ocorrência, mudanças no estado de coisas do mundo
  - Têm influência na representação
  - Várias propostas e nomenclaturas, desde Aristóteles



## Eventos

- Tipos de eventos

- Vendler (1967), Dowty (1979)

- **Stative:** *I know my departure gate.*
  - Eventos passam noção de propriedade, aspecto ou estado em um determinado tempo
- **Activity:** *John is flying.*
  - Eventos em que não há um ponto de término em particular
- **Accomplishment:** *Sally booked her flight.*
  - Eventos que têm um ponto de término natural e resultam em um determinado estado
- **Achievement:** *She found her gate.*
  - Eventos similares aos de *accomplishment*, mas acontecendo em um determinado instante e não equacionados com alguma atividade em particular que leve a um estado

65

## Eventos

- Tipos de eventos

- Chafe (1979)/Borba (1996) e os tipos de verbos

- **Ação:** *Paulo correu.*
  - Verbos de ação expressam uma atividade realizada por um sujeito agente
- **Processo:** *O leite ferveu.*
  - Os verbos de processo expressam um evento ou sucessão de eventos que afetam um sujeito paciente ou experimentador
- **Ação-processo:** *João quebrou o copo.*
  - Como processo, o verbo implica uma mudança na condição de um nome, seu paciente; como ação, expressa o que alguém, seu agente, faz
- **Estado:** *Ele vive.*
  - Os verbos de estado expressam uma propriedade (estado, condição, situação) localizada no sujeito, que é, pois, mero suporte dessas propriedades ou, então, seu experimentador ou beneficiário

66

## Lógica de Descrição

- *Description Logics*
  - Consiste em subconjuntos da Lógica de 1ª ordem úteis e computacionalmente tratáveis
  - Arcabouço conceitual para modelagem de certos domínios
  - Embasa a Web Semântica
    - Ontologias e OWL (**Web Ontology Language**)

67