Outros Métodos de Procura

Exemplos

- Jogo dos oito :-)
- Mundo dos blocos (ex: torre de Hanoi)
- Poblema das rainhas
- Criptoaritmética
- Missionários e Canibais
- Resta-um
- e muitos outros...

Rainhas

- 2 formas de se resolver o problema: incremental e completo
- Possíveis estados e jogadas:

1. – a) qq arranjo de 0 a 8 rainhas no tabuleiro
   – b) adicionar 1 rainha a qq posição do tabuleiro
2. – a) arranjos de 0 a 8 rainhas com nenhuma em posição de ataque
   – b) colocar 1 rainha na próxima coluna vazia mais à esquerda de forma que não seja atacada por outra rainha
3. – a) arranjos de 8 rainhas, 1 em cada coluna
   – b) mover qq rainha atacada para outra posição na mesma coluna

Criptoaritmética

```
                 SOLUCAO
  FORTY          29786    com: F = 2
+   TEN            850         O = 9
+   TEN            850         R = 7 etc
--------         ------
  FIFTY          31486
```

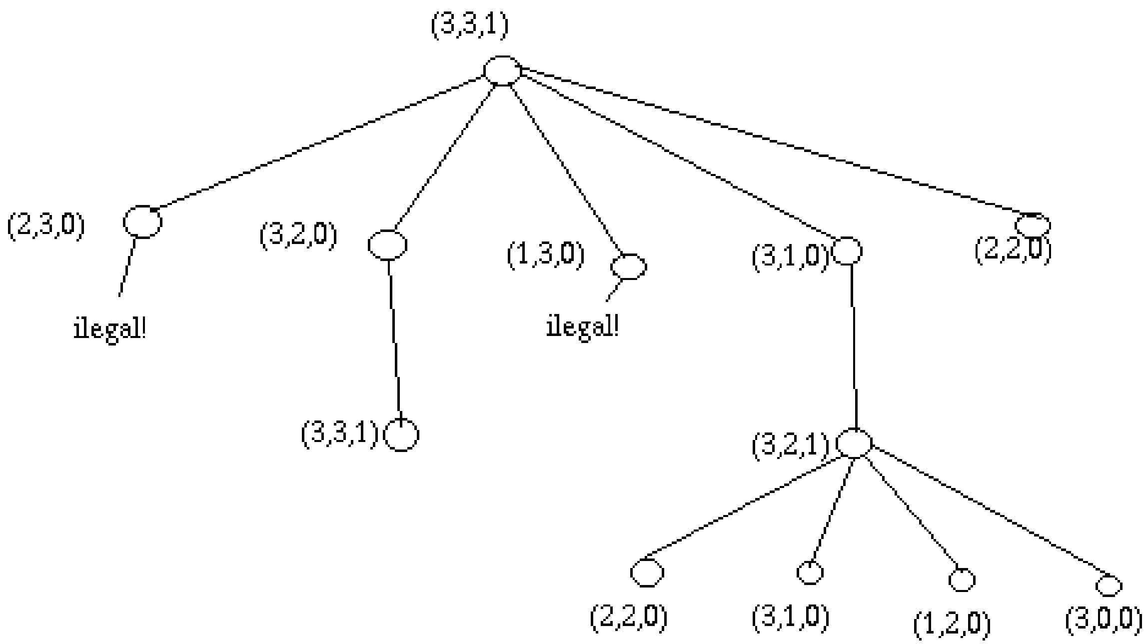
Criptoaritmética

- estados: conjunto de letras que são substituídas por dígitos
- operadores: substituir todas as ocorrências de uma letra por um dígito que ainda não tenha aparecido
- possíveis regras de seleção dos dígitos: ordem alafbética para evitar repetições, seleção mais restrita respeitando as operações aritméticas
- estado final: jogo contém somente dígitos e representa uma soma correta
- custo da solução: zero

Missionários e Canibais

- 3 missionários e 3 canibais estão na margem de um rio com um bote onde cabem, no máximo, 2 pessoas.
- Problema: encontrar uma forma de passar todos para a outra margem sem nunca deixar um grupo de missionários em uma margem com um número maior de canibais.
- estados: qq seqüência ordenada de 3 números representando o no. de missionários, canibais e botes na margem do rio.
- estado inicial: (3,3,1)
- estado final: (0,0,0)
- custo: número de travessias no rio.
- regras: tirar 1 missionário, tirar 1 canibal, tirar 2 canibais etc...

Missionários e Canibais



Sistemas de Produção

- Elementos principais de um sistema de produção em IA:
  - bancos de dados global
  - regras de produção ou restrições
  - sistema de controle

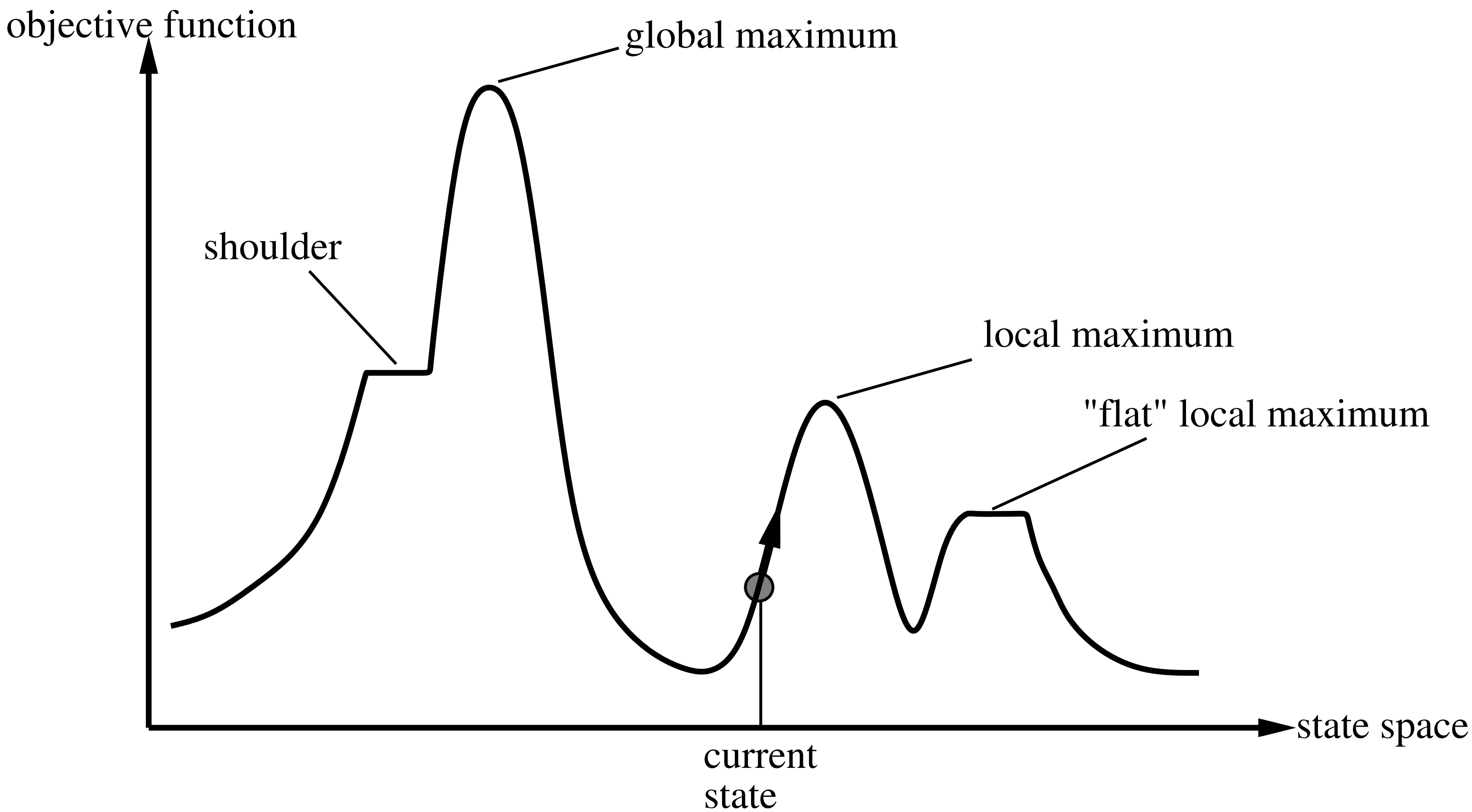
Estratégias de controle

- irrevogáveis: nunca retornam por um caminho já explorado
- tentativa: "backtracking" (métodos não informados e informados).

Algoritmos de Melhoramento Iterativo

- Utilizados em problemas cuja descrição já contém toda a informação para encontrar a solução (ex: n-rainhas e layout de circuitos VLSI)
- parte-se de uma config inicial conhecida e tenta-se melhorar a solução.

Algoritmos de Melhoramento Iterativo



Algoritmos de Melhoramento Iterativo

- Exemplos:
  - Método Irrevogável: Hill-Climbing (busca de máximo/mínimo local)
  - Métodos de tentativa: (busca de máximo/mínimo global) podem temporariamente tornar estados piores.
    - random restart-hill-climbing
    - simulated annealing
    - busca tabu
    - busca paralela

Hill Climbing (ou gradiente descendente)

- tenta fazer modificações que melhorem o estado corrente
- 2 desvantagens:
  - máximos locais
  - platôs: percorre estados aleatoriamente porque a função de avaliação não muda muito
- Exemplo: Jogo dos oito :-)

Hill Climbing: exemplo

```
2 8 3    2 8 3    2   3     2 3    1 2 3    1 2 3
1 6 4    1   4    1 8 4   1 8 4      8 4    8   4
7   5    7 6 5    7 6 5   7 6 5    7 6 5    7 6 5

f=-4     f=-3     f=-3    f=-2     f=-1     f=0
```

- No caso de não se poder aplicar a regra, o processo termina.
- Solução = máximo local, hill-climbing não encontra máximo global.

Hill Climbing: exemplo

- não funciona para:

```
1 2 5       1 2 3
  7 4  =>     7 4
8 6 3       8 6 5


f=-2
```

- Qq movimento diminui o valor da função de avaliação.

Random Restart Hill Climbing

- executa uma série de buscas hill-climbing a partir de estados iniciais aleatórios
- cada um roda até terminar ou até não ter nenhum progresso
- salva o melhor resultado obtido até então
- pode ter no. finito de iterações ou continuar até não conseguir melhorar o melhor valor encontrado
- se superfície de busca contém muitos máximos locais, busca exponencial
- geralmente, solução boa pode ser encontrada em um número pequeno de iterações.

Simulated Annealing

- Invés de começar novamente aleatoriamente quando passa num máximo local, permite que a busca escape do máximo local "descendo a montanha".

Simulated Annealing

```
function SA(problem,schedule) return a solution state
   current <- MAKE_NODE(INITIAL_STATE[problem])
   for t <- 1 to infinito do
       T <- schedule(t)
       if T = 0 then return current
       next <- sucessor de current selecionado
               aleatoriamente
       deltaE <- value[next] - value[current]
       if deltaE > 0 then current <- next
       else current <- next with prob e^(deltaE/T)
       endif
   endfor
```

Simulated Annealing

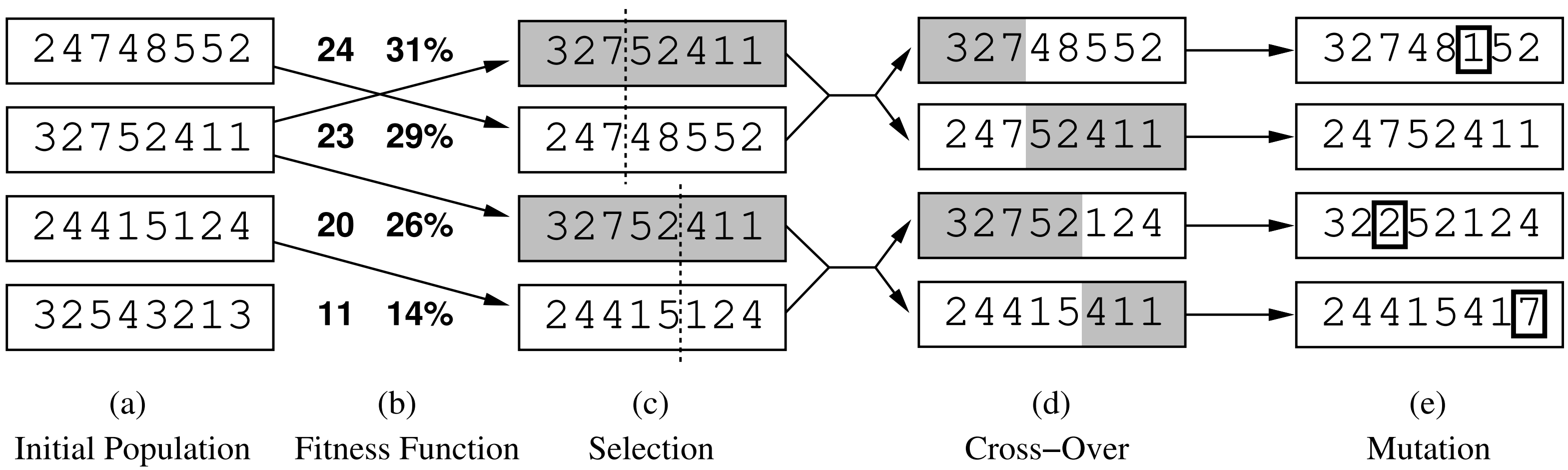
```
P = e^(deltaE/T)
n = sorteio de um no. de 0 a 1
if n < P then current <- next
```

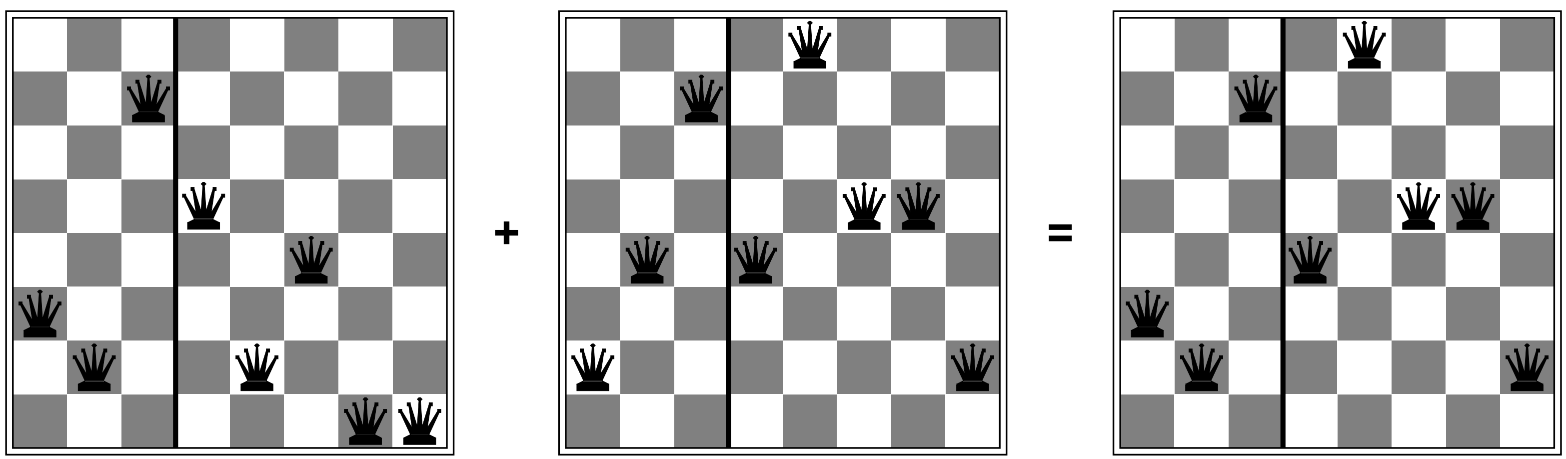
- ou seja: quanto maior a prob mais chance de aceitar mover para passos de custo pior.

Outros Algoritmos

- Algoritmos Genéticos
  - Operações: "crossover", mutação e reprodução
  - começa de uma população inicial
  - aplica as operações
  - calcula uma função de "fitness" para cada indivíduo da população
  - pode eliminar indivíduos menos "fit"

Algoritmos Genéticos





Outros Algoritmos

- Satisfação de Restrições
  - tipo especial de problema que satisfaz propriedades estruturais além dos requisitos básicos para problemas gerais de busca
  - estados: conjunto de variáveis
  - domínio: conjunto possível de valores que uma variável pode assumir (discreto/contínuo, finito/infinito)
  - estado inicial: todas as variáveis com valores possíveis iniciais
  - estado final: valores finais para as variáveis que respeitem as restrições do problema
  - profundidade máxima da árvore: número de variáveis

Satisfação de Restrições: Exemplo

- n-rainhas
  - variáveis: posições no tabuleiro
  - restrições: nenhuma rainha pode atacar outra na mesma linha, diagonal ou coluna
  - valores iniciais das variáveis: V_1 in $1..n$, V_2 in $1..n$ etc, com n largura do tabuleiro

Satisfação de Restrições: Possíveis algoritmos

- aloca uma nova rainha para uma nova coluna a cada nível (solução incremental)
- complexidade: $n^n =\approx \pi D_i = D_1 \times D_2 \times ... \times D_n$
- fator de ramificação: n
- fator máximo de ramificação: $n \times n = \sum D_i = D_1 + D_2 + ... + D_n$
- se todas as variáveis fossem instanciadas com todos os valores possíveis no primeiro nível da árvore

Satisfação de Restrições: Possíveis algoritmos

```
n = 8
                    (0,0,0,0,0,0,0,0)


N  (1,0,0,0,0,0,0,0)  (0,1,0,0,0,0,0,0)  (0,0,1,0,0,0,0,0) ....
i  (2,0,0,0,0,0,0,0)  (0,2,0,0,0,0,0,0)  (0,0,2,0,0,0,0,0) ....
v   ..........
1  (8,0,0,0,0,0,0,0)  (0,8,0,0,0,0,0,0)  (0,0,8,0,0,0,0,0) ....

N  (1,1,0,0,0,0,0,0)  (0,1,1,0,0,0,0,0)  (1,0,1,0,0,0,0,0) ....
i  (2,2,0,0,0,0,0,0)  (0,2,2,0,0,0,0,0)  (2,0,2,0,0,0,0,0) ....
v   ..........
2  (8,8,0,0,0,0,0,0)  (0,8,8,0,0,0,0,0)  (8,0,8,0,0,0,0,0) ....
```

Satisfação de Restrições: Possíveis algoritmos

- utilizando BP a sub-árvore que contém

  `(0,0,0,0,0,0,0,0) -> (1,0,0,0,0,0,0,0) ->`

  `(1,1,0,0,0,0,0,0) ->`

  `(1,1,1,0,0,0,0,0) -> (1,1,1,1,0,0,0,0) -> ...`

- será explorada mesmo sabendo que (1,1,1,1,1,1,1,1) não é solução

Satisfação de Restrições: Possíveis algoritmos

- solução: colocar o teste de restrição a cada rainha colocada no tabuleiro
- tb não é a melhor solução!
- Suponha que já conseguimos alocar 6 rainhas, mas esta alocação ataca a oitava rainha.
- BP testa todas as possibilidades de colocação da sétima rainha!

Satisfação de Restrições: Possíveis algoritmos

- solução: algoritmos *Forward Checking* e *Lookahead* ou simplesmente "consistência de arcos"
- Forward Checking: retira dos domínios de outras variáveis todos os valores impossíveis que violam as restrições
- Lookahead: além de executar forward checking, verifica se o domínio modificado de cada variável conflita com os outros.

Forward Checking: Exemplo

```
n = 8
(1) V1 = 1  ==> V2 = {3,4,5,6,7,8}
                V3 = {2,4,5,6,7,8}
                V4 = {2,3,5,6,7,8}
                V5 = {2,3,4,6,7,8}
                V6 = {2,3,4,5,7,8}
                V7 = {2,3,4,5,6,8}
                V8 = {2,3,4,5,6,7}

(2) V2 = 3  ==> V3 = {5,6,7,8}
                V4 = {2,6,7,8}
                V5 = {2,4,7,8}
                V6 = {2,4,5,8}
                V7 = {2,4,5,6}
                V8 = {2,4,5,6,7}
```

```
                      +---+---+---+---+---+---+---+---+
                      | X |   |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
                      |   |   |   |   |   |   |   |   |
n-rainhas             +---+---+---+---+---+---+---+---+
n=8                   |   | X |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
                      |   |   |   |   |   |   |   |   |
apos V1=1             +---+---+---+---+---+---+---+---+
     V2=3             |   |   |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
                      |   |   |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
                      |   |   |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
                      |   |   |   |   |   |   |   |   |
                      +---+---+---+---+---+---+---+---+
```

Satisfação de Restrições: Possíveis algoritmos

- para domínios infinitos: programação linear, simplex, revised simplex, convex hull, eliminação de Gauss.
- para domínios finitos: forward checking, lookahead, consistência de arcos em geral.
- Para domínios finitos, dois problemas:
  - escolha da *variável*:
    - *most-constrained*: menor domínio
    - *most constraining*: restringe ao máximo os domínios das outras variáveis
  - escolha do *valor* da variável:
    - princípio *first-fail*.
    - *least constraining*: valor que afeta menos o conjunto de valores das outras variáveis

```
Coloracao de Mapas: 3 cores  +------------+
azul                         |B           |
verde              +---------+            |
vermelho           |A        |            |
                   |         |            |
          +--------+--+      |            |
          |C          +------+------+     |
          |           |             |     |
          +-+----+----+             |     |
            |    |                  +---+-+
            |    |E                 |   |
            |    +------------------+   +---------+
            |                           |         |
            |                           |         |
            |                           |         |
            |F                          |D        |
            +---------------------------+---------+
```

Satisfação de Restrições

- variável most-constrained: permite resolver n-rainhas com n igual a 100.
- forward checking puro: só consegue resolver até 30.
- valor least-constraining: permite resolver n-rainhas com n igual a 1000.

Apoio à Decisão: Lidando com Incertezas

Incerteza

- Falta de informação suficiente.
- Conhecimento não completo ou não correto.
- Planos condicionais podem lidar com incerteza de forma limitada.
- Ex: Plano para chegar ao aeroporto saindo 90 mins adiantado.
- Plano vai levar o passageiro ao aeroporto a tempo se:
  - o carro não quebrar
  - não faltar gasolina
  - não acontecer um acidente na estrada
  - ou com o próprio carro
  - não houver um terremoto...
- **Decisão Racional**: depende da importância relativa de vários objetivos e da probabilidade com q eles podem ser alcançados.

Lidando com Incerteza

- Exemplo simples: diagnósticos.
- Para odontologia:
  $\forall p \; Sintoma(p, DorDeDente) \Rightarrow Enfermidade(p, Carie)$: regra errada.
- Alternativa:
  $\forall p \; Sintoma(p, DorDeDente) \Rightarrow Enfermidade(p, Carie) \vee Enfermidade(p, Gengivite) \vee Enfermidade(p, SisoIncluso)...$
- Disjunção pode ser ilimitada.
- Outra alternativa, regra causal:
  $\forall p \; Enfermidade(p, Carie) \Rightarrow Sintoma(p, DorDeDente)$: regra errada.
- Conclusão: cáries e dores de dentes não estão necessariamente relacionados.

Lidando com Incerteza

- Como representar alguma relação? Graus de incerteza ou de credibilidade (*degrees of belief* ou *degrees of truth*).
- Lógica de primeira ordem não consegue representar incerteza.
- 3 razões:
  - Laziness: muito trabalhoso codificar listas completas de antecedentes ou consequentes necessários para representar todas as regras, e muito ineficiente utilizar o conjunto enorme de regras.
  - Ignorância teórica: ciência médica (pe) não tem uma teoria completa sobre o domínio.
  - Ignorância prática: mesmo sabendo todas as regras, pode aparecer um paciente com dados novos q não estavam previstos.

Lidando com Incerteza

- Para lidar com incerteza: teoria das probabilidades: degree of belief, lógica fuzzy: degree of truth.
- Probabilidade fornece uma forma de resumir a incerteza proveniente de 'laziness' e ignorância.
- Pe, *80% de pacientes já tratados que têm dor de dente, têm cárie*. 80% compreende:
  - todos os casos em que reunimos todas as condições necessárias para concluir que se um paciente tem dor de dente, tem cárie,
  - todos os casos em que o paciente tem dor de dente e cárie, mas os dois não estão relacionados.

Lidando com Incerteza

- 20% compreende todas as possíveis causa de dor de dente não relacionadas por laziness ou ignorância.
- Prob = 0, sentença é categoricamente falsa.
- Prob = 1, sentença é categoricamente verdadeira.
- **evidência**: utilizada pelo agente para atribuir novas probabilidades às proposições a partir de percepções do mundo.
- **probabilidade incondicional**: inicial.
- **probabilidade condicional**: posterior, após evidências serem obtidas.

Incerteza e Decisões Racionais

- A_{90} ou A_{120} ou A_{1440}? O q é mais racional?
- Depende dos objetivos e do tempo q uma pessoa está preparada para esperar pelo vôo no aeroporto.
- **Preferências** diferentes para as saídas do plano.
- Pe, Uma saída de um plano pode conter fatores tais como "agente vai chegar a tempo", e "tempo de espera no aeroporto".
- *Utility theory*: para atribuir graus de "utilidade" de um determinado estado.

Agente baseado em Teoria da Decisão

- **Teoria da Decisão** = teoria das probabilidades + Teoria da Utilidade.
- Idéia fundamental: agente é racional sss escolher uma ação que produza o caminho mais útil, em relação a todos os possíveis resultados da ação (princípio MEU, Maximum Expected Utility).
- Agente baseado em teoria da decisão: estrutura similar ao agente lógico.
- Linguagem para representação: lógica proposicional com probabilidades. Diferença sintática entre probabilidade incondicional e condicional (como em probabilidade).

```
function DT-AGENT(percept) returns an action
   static: a set probabilistic beliefs about the state of the world

   calculate updated probabilities for current state based on
       available evidence including current percept and previous action
   calculate outcome probabilities for actions,
       given action descriptions and probabilities of current states
   select action with highest expected utility
       given probabilities of outcomes and utility information
   return action
```

Revisão Probabilidade

- Probabilidade incondicional: P(A), probabilidade incondicional de que proposição A é verdadeira.
- Pe, P(Cárie) = 0.1, 10% de chance de que o paciente tem cárie, *na ausência de qq outra informação.*
- Variáveis aleatórias (ref. prob.):
  - P(Tempo=ensolarado) = 0.7
  - P(Tempo=nublado) = 0.08
- Domínio: possíveis valores $\langle x_1, x_2, \ldots, x_n \rangle$ de uma variável aleatória (conjunto discreto para exemplos).
- Convenção: A, B... para variáveis booleanas. X, Y,... para variáveis com vários valores.
- **P**(Tempo) = (0.7,0.2,0.08,0.02): distribuição de probabilidade para a variável Tempo.

Revisão Probabilidade

- P(a|b): probabilidade condicional.
- P.ex., P(carie|dorDeDente) = 0.8: chance de 80% do paciente ter cárie, dado que "tudo que sabemos até agora" é que ele tem dor de dente.
- $P(a \mid b) = \frac{P(a \wedge b)}{P(b)}, com P(b) > 0$
- Regra do produto: $P(a \wedge b) = P(a \mid b)P(b)$
- $\mathbf{P}(\mathrm{X,Y}) = \mathbf{P}(\mathrm{X|Y})\mathbf{P}(\mathrm{Y})$
- Nem sempre evidência está disponível no BD: inferência probabilística.

Axiomas da Probabilidade

- $0 \leq P(a) \leq 1$
- $P(true) = 1$, $P(false) = 0$
- $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$ (ref. diag Venn).
- Propriedades: (com $b = \neg a$)

$$
\begin{aligned}
P(a \vee \neg a) = & \; P(a) + P(\neg a) - P(a \wedge \neg a) \\
P(True) = & \; P(a) + P(\neg a) - p(false) \\
1 = & \; P(a) + P(\neg a) \\
p(\neg a) = & \; 1 - P(a)
\end{aligned}
$$

- de Finetti: Crenças inconsistentes levam agente sempre a perder ou produzir ações não desejadas. Axiomas são importantes, pq não deixam agentes criarem inconsistências.

Exemplo

Agent 1		Agent 2		Outcome for Agent 1			
Proposition	Belief	Bet	Stakes	$A \wedge B$	$A \wedge \neg B$	$\neg A \wedge B$	$\neg A \wedge \neg B$
A	0.4	A	4 to 6	-6	-6	4	4
B	0.3	B	3 to 7	-7	3	-7	3
$A \vee B$	0.8	$\neg(A \vee B)$	2 to 8	2	2	2	-8
				-11	-1	-1	-1

Agente 1 acredita na primeira coluna da tabela e em $P(a \wedge b) = 0$.

Distribuição de Probabilidade Conjunta

- $\mathbf{P}(X_1, \ldots, X_n)$: atribui probabilidades a todos os possíveis *eventos atômicos*.
- *Evento Atômico*: atribuição de valores particulares para todas as variáveis, especificação completa do domínio.
- Pe:

	DorDeDente	¬ DorDeDente
Carie	0.04	0.06
¬ Carie	0.01	0.89

Utilização de Distribuição de Probabilidade Conjunta

- P(carie): 0.06 + 0.04 = 0.10 (Marginal probability)
- P(carie $\vee$ dorDeDente) = 0.04 + 0.01 + 0.06 = 0.11
- Mecanismo de inferência pode encontrar probabilidades de uma proposição A, dada a evidência B.
- P(carie| dorDeDente) = $\frac{0.04}{0.04+0.01}$ = 0.80
- Em problemas reais, não é aconselhável usar distribuição de probabilidade conjunta: não prático armazenar a tabela com 2^n entradas para n variáveis booleanas.
- sistemas de raciocínio modernos: lidam diretamente com probabilidades condicionais.

Regra de Bayes

- $P(B \mid A) = \frac{P(A|B)P(B)}{P(A)}$: teorema, lei ou regra de Bayes.
- Fundamenta a maior parte dos sistemas probabilísticos de inferência.
- Forma geral: $\mathbf{P}(\mathrm{Y}|\mathrm{X}) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)}$
- Na prática, regra de Bayes é útil, porque na maioria dos problemas temos estimativas para as duas probabilidades incondicionais e para a probabilidade condicional.
- P.ex., um médico sabe que meningite causa enrijecimento nos músculos do pescoço do paciente, em 50% dos casos.
- Tb sabe dois fatos incondicionais: a prob de um paciente ter meningite é $\frac{1}{50.000}$, e a prob de um paciente ter enrijecimento dos músculos do pescoço é $\frac{1}{20}$.

Regra de Bayes

- S = paciente tem enrijecimento do pescoço, M = paciente tem meningite.

$$
\begin{aligned}
P(S \mid M) &= 0.5 \\
P(M) &= \tfrac{1}{50.000} \\
P(S) &= \tfrac{1}{20} \\
P(M \mid S) &= \tfrac{P(S|M)P(M)}{P(S)} = \tfrac{0.5 \times \frac{1}{50.000}}{\frac{1}{20}} = 0.0002
\end{aligned}
$$

- ie, somente 1 em cada 5000 pacientes com enrijecimento do pescoço tem meningite.

Regra de Bayes

- conhecimento sobre diagnósticos é mais superficial do que conhecimento causal.
- se surgir uma epidemia de meningite, P(M) vai subir.
- um médico que não usa regra de Bayes não vai saber como atualizar seus dados. Um médico usando regra de Bayes sabe que P(M| S) cresce proporcionalmente com P(M).

Regra de Bayes – Normalização

- Considere novamente a equação para calcular a probabilidade de meningite dado enrijecimento do pescoço:
  $P(M \mid S) = \frac{P(S|M)P(M)}{P(S)}$
- Considere a possibilidade do paciente ter enrijecimento do pescoço e isto causar uma distensão dos músculos:
  $P(W \mid S) = \frac{P(S|W)P(W)}{P(S)}$
- **probabilidade relativa**: $(P(S \mid W) = 0.8$ e $P(W) = \frac{1}{1000})$.
  $\frac{P(M|S)}{P(W|S)} = \frac{P(S|M)P(M)}{P(S|W)P(W)} = \frac{1}{80}$
- ie, distensão é 80 vezes mais frequente do que meningite, dado que o paciente tem enrijecimento dos músculos do pescoço.

Regra de Bayes – Normalização

- Em alguns casos, probabilidades relativas são suficientes para tomar decisões, mas em outros casos, há necessidade de calcular números mais precisos, sem precisar utilizar P(S) (probabilidade incondicional): **normalização**.

  $P(M \mid S) = \frac{P(S|M)P(M)}{P(S)}$

  $P(\neg M \mid S) = \frac{P(S|\neg M)P(\neg M)}{P(S)}$

  Adicionando: (obs: $P(M \mid S) + P(\neg M \mid S) = 1$)

  $P(S) = P(S \mid M)P(M) + P(S \mid \neg M)P(\neg M)$

- Substituindo na regra de Bayes:

  $P(M \mid S) = \frac{P(S|M)P(M)}{P(S|M)P(M)+P(S|\neg M)P(\neg M)}$

- regra geral: $\mathbf{P}(Y \mid X) = \alpha \mathbf{P}(X \mid Y)\mathbf{P}(Y)$

Usando a Regra de Bayes: combinação de evidências

- Assuma:

  $P(Carie \mid DorDeDente) = 0.8$

  $P(Carie \mid MotorPrendeu) = 0.95$

- O que um(a) dentista pode concluir se o motor prendeu no dente em que o paciente sente dor?

- Usando tabela de distr. de prob. conjunta, bastaria consultar tabela para encontrar $P(Carie \mid DorDeDente \wedge MotorPrendeu)$

- Usando Bayes:

  $P(Carie \mid DorDeDente \wedge MotorPrendeu) = \frac{P(DorDeDente \wedge MotorPrendeu | Carie) P(Carie)}{P(DorDeDente \wedge MotorPrendeu)}$

Usando a Regra de Bayes: combinação de evidências

- esta forma pode levar a um número exponencial de valores de probabilidade se tivermos conjunções com mais variáveis. Por que não voltar a usar tabela de prob. conjunta neste caso?
- Polêmica entre pesquisadores que decidiram adotar métodos aproximados para tratar combinações de evidências, invés de teoria das probabilidades.

Usando a Regra de Bayes: combinação de evidências

- Em alguns domínios, Bayes pode ser simplificada e usar menos valores de probabilidades para produzir resultados: **atualização bayesiana** (*bayesian updating*).

$$P(Carie \mid DorDeDente) = P(Carie)\frac{P(DorDeDente \mid Carie)}{P(DorDeDente)}$$

- qdo MotorPrendeu é observado, aplicamos novamente Bayes com DorDeDente como variável condicional de contexto:
  $P(Carie \mid DorDeDente \wedge MotorPrendeu) =$
  $P(Carie \mid DorDeDente)\frac{P(MotorPrendeu \mid DorDeDente \wedge Carie)}{P(MotorPrendeu \mid DorDeDente)}$

- Em atualização bayesiana, cada vez que uma nova evidência é observada, a crença é multiplicada por um fator que depende da nova evidência.

Usando a Regra de Bayes: combinação de evidências

- Ainda está complicado!
- $P(MotorPrendeu \mid DorDeDente \wedge Carie)$ não é mais fácil de ser calculado do que $P(DorDeDente \wedge MotorPrendeu \mid Carie)$!
- observação chave: cárie é causa direta da dor de dente e do motor ter agarrado ao dente, neste exemplo.
- Simplificação: **independência condicional** de DorDeDente e de MotorPrendeu, dado Carie:

$$P(MotorPrendeu \mid Carie \wedge DorDeDente) = P(MotorPrendeu \mid Carie)$$
$$P(DorDeDente \mid Carie \wedge MotorPrendeu) = P(DorDeDente \mid Carie)$$

- Simplificando:

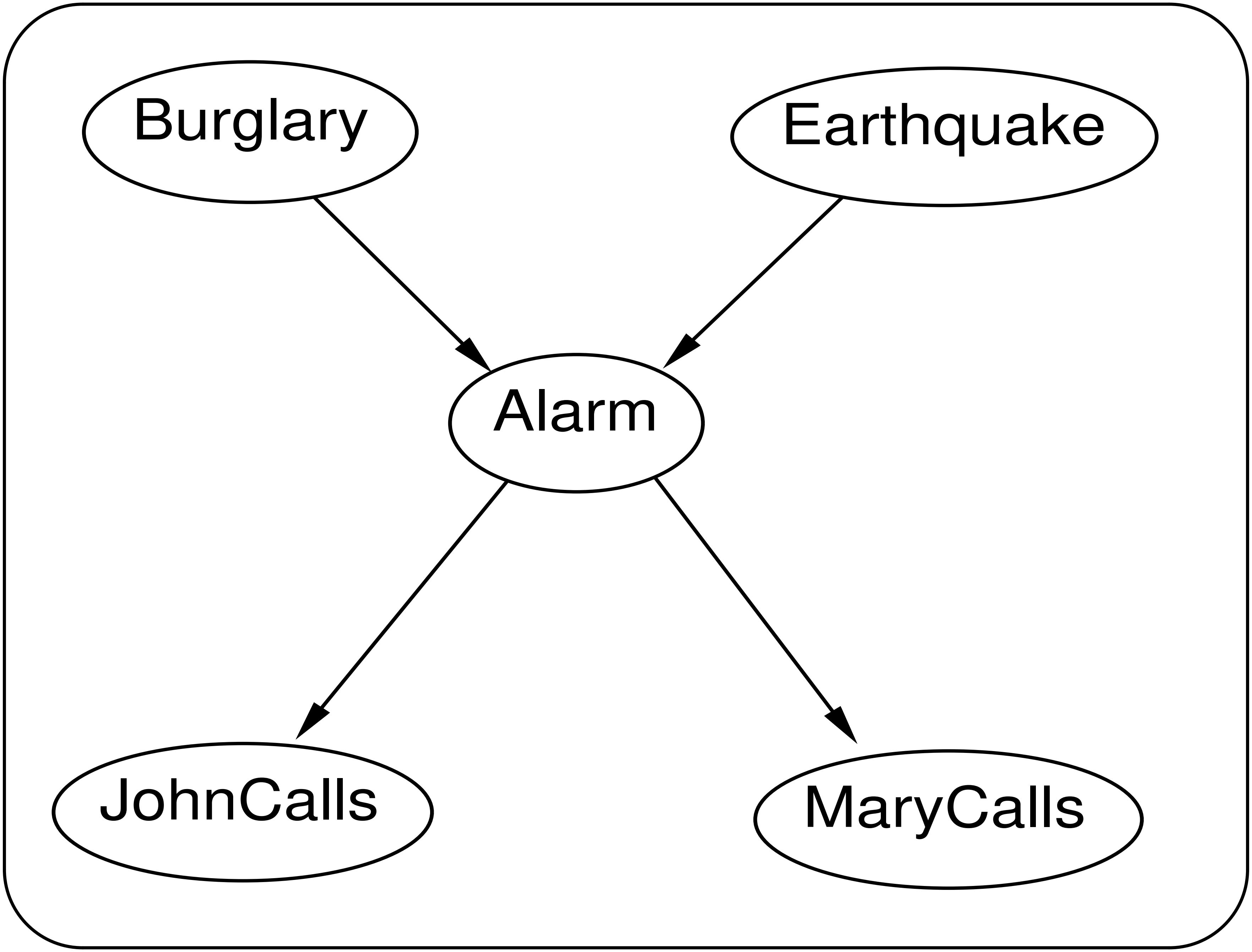
$$P(Carie \mid DorDeDente \wedge MotorPrendeu) =$$
$$P(Carie)\frac{P(DorDeDente|Carie)}{P(DorDeDente)}\frac{P(MotorPrendeu|Carie)}{P(MotorPrendeu|DorDeDente)}$$

Sistemas de Raciocínio Probabilístico

- Como construir sistemas de raciocínio utilizando modelos de redes e que usem incerteza de acordo com a teoria das probabilidades?
- **Rede de Crença** ou rede bayesiana: grafo com as seguintes características:
  - Nós representam variáveis aleatórias.
  - Arcos direcionados representam ligações diretas entre variáveis aleatórias.
  - Cada nó tem uma tabela de prob. cond. que quantifica o efeito dos pais deste nó.
  - O grafo não possui ciclos (DAG).
- relativamente fácil para o expert definir as relações do que definir as probs.

Sistemas de Raciocínio Probabilístico

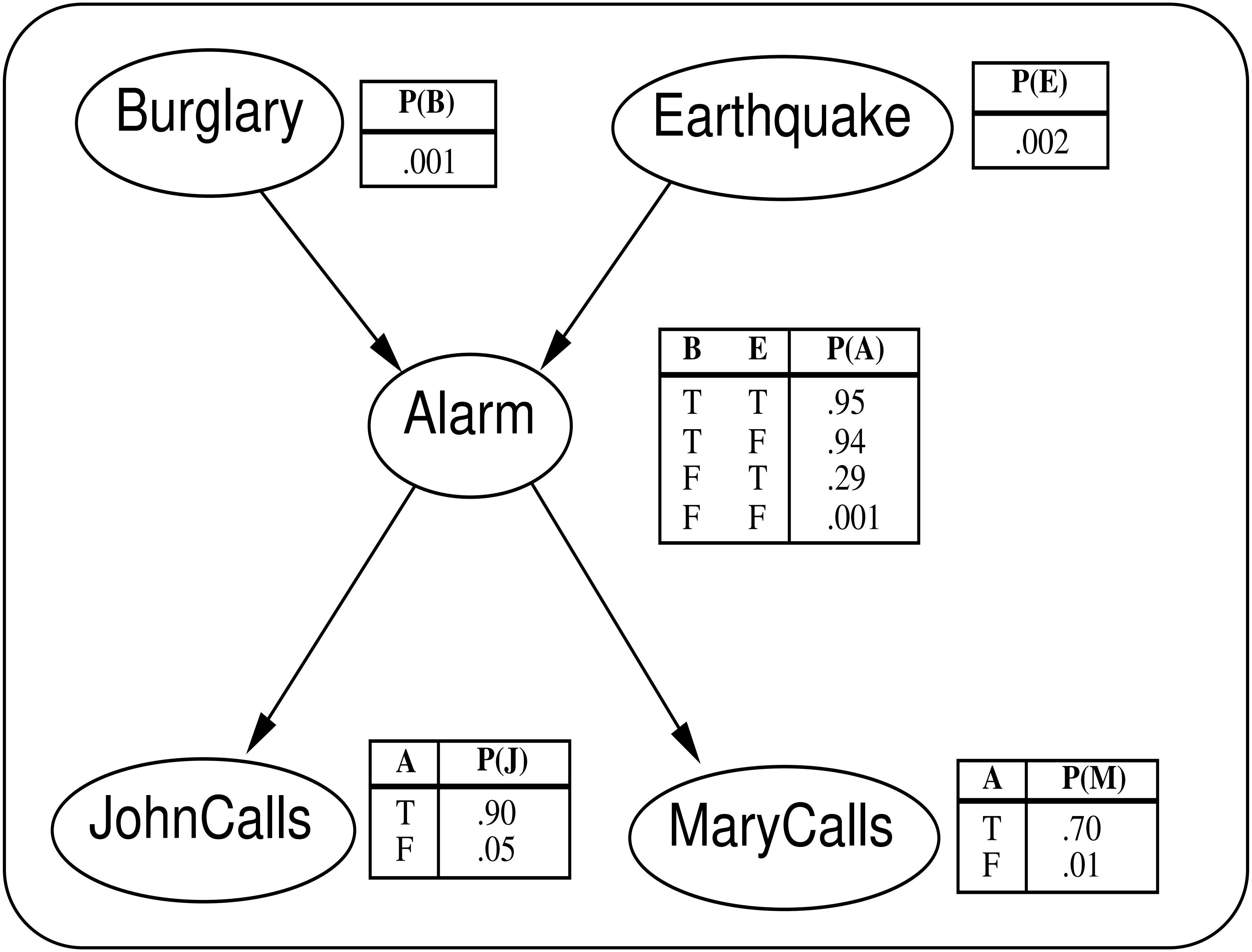
- Exemplo: alarme contra roubo.
- Alarme toca em duas situações: tentativa de roubo e terremoto.
- John e Mary são os vizinhos que avisam ao dono da casa se o alarme estiver tocando.
- John liga toda vez q o alarme toca e tb qdo o tel toca.
- Mary somente liga qdo o alarme toca, mas não ouve algumas vezes.
- Dada a evidência de quem ligou para o dono da casa, queremos descobrir a probabilidade de ter havido roubo.


Burglary
Earthquake
Alarm
JohnCalls
MaryCalls

Sistemas de Raciocínio Probabilístico

- Rede somente representa ligações diretas, causais.
- Nada é informado sobre Mary ouvir música alta ou de John confundir o tel com o alarme.
- Tabela de probabilidades condicionais:

Roubo	Terr	$\mathbf{P}(Alarme \mid Roubo, Terr)$
T	T	0.950 0.050
T	F	0.950 0.050
F	T	0.290 0.710
F	F	0.001 0.999


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
T T .95
T F .94
F T .29
F F .001
JohnCalls
A P(J)
T .90
F .05
MaryCalls
A P(M)
T .70
F .01

Semântica das Redes de Crenças

- Duas formas de entender:
  - representação da distribuição de probabilidade conjunta. Útil para **construir** redes.
  - conjunto de sentenças condicionalmente independentes. Útil para projetar procedimentos de inferência.
- Representando distribuição de probabilidade conjunta:
  - cada entrada na tabela pode ser calculada através da info na rede.
  - uma entrada genérica representa a probabilidade de uma conjunção de valores atribuídos a cada variável: $P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$.
    $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Pais(X_i))$

Construção de Redes de Crenças

- cada entrada na tabela é representada pelo produto dos elementos apropriados das tabelas de probabilidades condicionais (TPCs).
- TPCs fornecem uma representação decomposta da distribuição conjunta.
- Exemplo:

  $P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = P(J \mid A)P(M \mid A)P(A \mid \neg B \wedge \neg E)P(\neg B)P(\neg E) = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062$
- Métodos melhores do que obter a tabela inteira de distribuição de probabilidade conjunta.

Construção de Redes de Crenças

- Método para construir redes de crenças:
  - Rede é construída de forma que cada nó é condicionalmente independente dos seus predecessores, dada a probabilidade dos seus pais.
  - equação $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Pais(x_i))$ usada para guiar o engenheiro de conhecimento a construir a topologia da rede.
  - Para construir uma rede de forma que esta tenha a estrutura correta para o domínio, escolhe-se nós pais adequados para garantir que cada nó é condicionalmente independente de seus antecessores.

Construção de Redes de Crenças

- Em geral:

  $\mathbf{P}(X_i \mid X_{i-1}, \ldots, X_1) = P(X_i \mid Pais(X_i))$, desde que $Pais(X_i)$ seja subconjunto ou igual a $\{x_{i-1}, \ldots, x_1\}$

- Esta condição é alcançada desde que se rotule os nós da rede em qq ordem q seja consistente com a ordem parcial implícita na estrutura do grafo.

- Ex: Mary telefona: não é **diretamente** influenciado por tentativa de assalto ou terremoto. É influenciado pelo efeito de tentativa de assalto ou terremoto, ou seja, soar o alarme.

- o Fato de John telefonar tb não tem influência direta sobre o fato de Mary telefonar. Neste caso temos independência condicional:

  $\mathbf{P}(MaryTel \mid JohnTel, Alarme, Terr, Assalto) =$

  $\mathbf{P}(MaryTel \mid Alarme)$

Construção de Redes de Crenças

- Procedimento geral para construção de redes:

1. Escolha o conj de variáveis X_i relevantes que descrevam o domínio.
2. Escolha ordem para as variáveis.
3. Eqto há vars:
   a) Pegue uma var X_i e adicione um nó na rede para X_i.
   b) Construir $Pais(X_i)$ com um conj mínimo de nós que já estejam na rede, tal que a prop de indep cond seja satisfeita.
   c) Defina a tabela de prob cond p/ X_i.

Construção de Redes de Crenças

- Procedimento garante que a rede é acíclica.
- Rede não contém valores de probabilidades redundantes.
- Garante que axiomas da prob não são violados.

Compactação e Ordenação de Nós

- Redes de crenças + **compactas** do que distribuição de prob conjunta.
- Sistemas **localmente estruturados** ou esparsos com info distribuída pelos nós.
- Crescimento polinomial.
- Em redes de crenças, podemos assumir que para a maioria dos domínios, cada variável aleatória é diretamente influenciada por no max k outras vars (nós pais).
- Qtde necessária de números para a TPC de cada nó: 2^k.
- Para a rede completa (n nós): $n2^k$.

Compactação e Ordenação de Nós

- Ex concreto: rede com 20 nós e no max 5 pais p/ cada nó:
  - redes de crença: 640 números.
  - tabela de distr de prob conj: ordem de 10^6 números.
- Número de links extra na rede = maior precisão, mas pode não se justificar devido ao aumento do tamanho das tabelas.
- Regra geral: adicionar à rede primeiro os nós causadores de algum efeito e depois seus efeitos.

Representação de TPCs

- Problema: escolher as probs condicionais das TPCs.
- Relação entre pais e filhos pode se encaixar numa distribuição canônica. Neste caso, probs podem ser especificadas através de nomes e talvez parms adicionais.
- Ex mais simples: nós determinísticos, probs são iguais as probs dos pais.
- Nós não determinísticos: relação *ruidosa* OU.
- Representação das probs:
  - Se todos os pais F, nó de saída F, com 100% de certeza.
  - Se apenas 1 dos pais é V, nó de saída é F com prob = parâmetro ruidoso daquele nó pai que é V.
- Ex: $P(Febre \mid Resf) = 0.4$, $P(Febre \mid Gripe) = 0.8$ e $P(Febre \mid Mal) = 0.9$

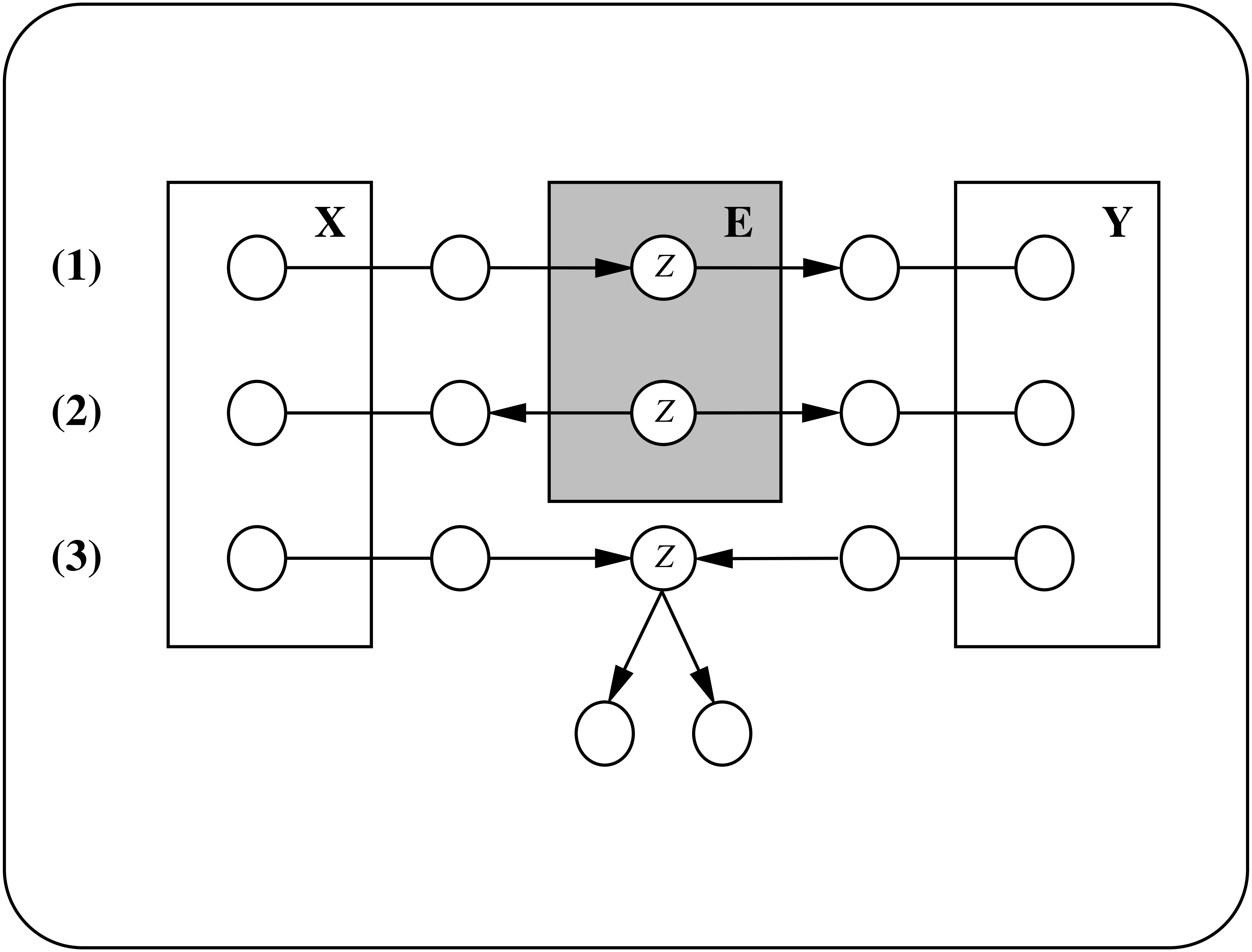
Resf	Gripe	Mal	P(Febre)	P(¬ Febre)
F	F	F	0.0	1.0
F	F	V	0.9	0.1
F	V	F	0.8	0.2
F	V	V	0.98	$0.02 = 0.2 \times 0.1$
V	F	F	0.4	0.6
V	F	V	0.94	$0.06 = 0.6 \times 0.1$
V	V	F	0.88	$0.12 = 0.6 \times 0.2$
V	V	V	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Relações de Independência Condicional

- Necessidade: saber se relações de independência condicional mais gerais (não somente de pais p/ filhos) existem para poder obter mecanismos de inferência capazes de responder consultas do tipo: "Existe algum conj de nós X independente de outro conj Y, dado conj de evidências E?"
- Método: **direction-dependent separation** ou **d-separation** (separação-d).
- separação-d: *se todo ramo não dirigido de um nó em X para um nó em Y é d-separado por E, então X e Y são condicionalmente independentes, dado E.*
- Um conj de nós E d-separa dois conj X e Y se todo ramo não dirigido de um nó em X para um nó em Y estiver **bloqueado**, dado E.

Relações de Independência Condicional

- Um ramo está bloqueado, dado um conj de nós E, se houver um nó Z no caminho tal que uma das três condições é satisfeita:
  1. Z está em E e Z tem um arco incidente e outro não incidente.
  2. Z está em E e Z tem ambos os arcos não incidentes.
  3. Nem Z nem nenhum descendente de Z estão em E, e ambos os arcos incidem em Z.


X
E
Y
(1)
(2)
(3)
Z
Z
Z

Inferência em Redes de Crenças

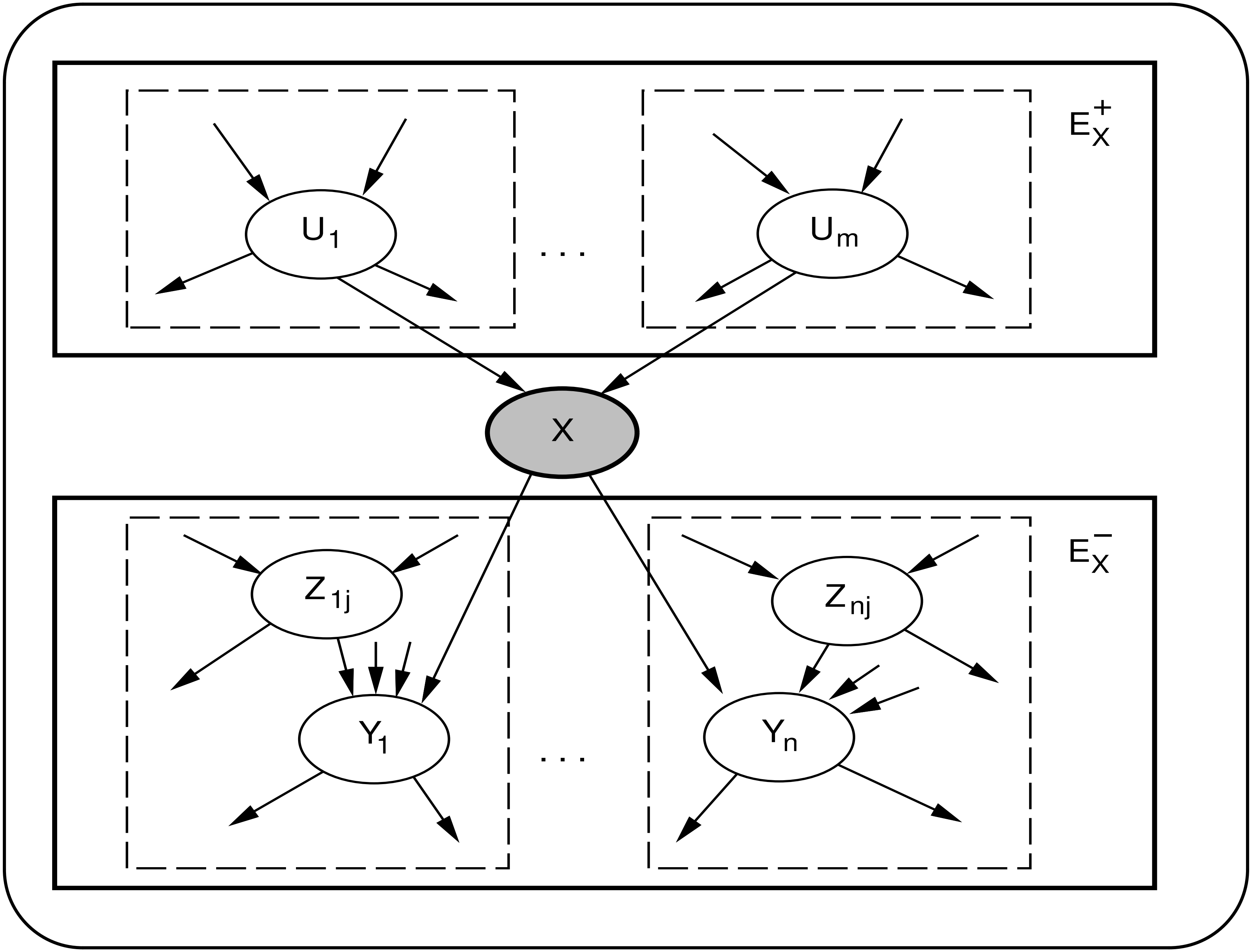
- Objetivo principal: computar distribuição de probabilidade posterior para um conj de variáveis de consulta, dados valores exatos para variáveis de evidência: $P(Cons \mid Evidencia)$.
- A princípio, qq nó pode servir como consulta ou evidência.
- Duas funções: BELIEF_NET_TELL p/ incluir novas evidências na rede e BELIEF_NET_ASK p/ computar novas probs p/ as variáveis de consulta.

Inferência em Redes de Crenças

- **Diagnósticas**: $P(Assalto \mid JohnTel)$ (efeitos p/ as causas).
- **Causais**: $P(JohnTel \mid Assalto)$ (causas p/ os efeitos).
- **Inter-causais**: $P(Assalto \mid Alarme \wedge Terr)$.
- **Mistas**: combinação de 1 ou mais dos casos acima.
- Redes de crenças ainda podem ser usadas para:
  - tomar decisões baseadas em probs na rede e em funções de utilidade do agente.
  - decidir quais variáveis de evidência observar para obter info mais útil.
  - fazer "análise de sensibilidade" para entender aspectos do modelo que têm > impacto nas probs das vars de consulta.
  - Explicar os resultados da inferência probabilística para o usuário.

Inferência em Redes de Crenças

- Algoritmo BELIEF_NET_ASK análogo ao backward-chain, faz inferências a partir das variáveis de consulta até encontrar alguma evidência.
- Algoritmo funciona somente para redes "singly connected", onde há no máximo um ramo não dirigido entre quaisquer dois nós da rede: **poli-árvores**.
- Algoritmos para redes mais gerais usam algoritmos poli-árvore como sub-rotinas.



E_X^+
U_1
. . .
U_m
X
E_X^-
Z_{1j}
Z_{nj}
Y_1
. . .
Y_n

Inferência em Redes de Crenças

- Nó X tem pais U e filhos Y.
- "singly connected" significa que todos os blocos são disjuntos e não têm links.
- X é a variável de consulta.
- Objetivo: computar $P(X \mid E)$.
- Conjunto de **suporte causal**: variáveis de evidência "acima" de X que estão conectadas a X através dos pais.
- Conjunto de **suporte evidencial**: variáveis de evidência "abaixo" de X que estão conectadas a X através dos filhos.
- $E_{U_i \mid X}$: evidências conectadas com todos os nós U_i, *exceto* via o ramo que passa por X.

Inferência em Redes de Crenças

- Estratégia geral:
  - Representar $\mathbf{P}(X \mid E)$ em termos de contribuições de E_X^+ e E_X^-.
  - Computar a contribuição de E_X^+ através do seu efeito nos pais de X. Obs: computar os efeitos dos pais de X em X pode ser feito de forma recursiva.
  - Computar a contribuição de E_X^- através do seu efeito nos pais de X. Obs: computar os efeitos dos pais de X em X pode ser feito de forma recursiva.
- Método: aplicar Bayes, outros métodos padrão de prob, simplificações (independência condicional).

Inferência em Redes de Crenças

- $\mathbf{P}(X \mid E) = \mathbf{P}(X \mid E_X^-, E_X^+) = \frac{\mathbf{P}(E_X^- | X, E_X^+)\mathbf{P}(X | E_X^+)}{\mathbf{P}(E_X^- | E_X^+)}$
- Como X d-separa E_X^+ de E_X^- na rede, podemos usar indep cond p/ simplificar primeiro termo do numerador. Tb podemos usar $\frac{1}{\mathbf{P}(E_X^- | E_X^+)}$ como cte de normalização:
- $\mathbf{P}(X \mid E) = \alpha \mathbf{P}(E_X^- \mid X)\mathbf{P}(X \mid E_X^+)$
- $\mathbf{P}(X \mid E_X^+) = \sum_u \mathbf{P}(X \mid \mathbf{u}, E_X^+) P(\mathbf{u} \mid E_X^+)$
- $\mathbf{P}(X \mid E_X^+) = \sum_u \mathbf{P}(X \mid \mathbf{u}) \prod_i \mathbf{P}(u_i \mid E_X^+)$
- $\mathbf{P}(X \mid E_X^+) = \sum_u \mathbf{P}(X \mid \mathbf{u}) \prod_i \mathbf{P}(u_i \mid E_{U_i | X})$

Inferência em Redes de Crenças

- Sejam Z_i pais de Y_i e z_i, um conj de valores para Z_i.

  $\mathbf{P}(E_X^- \mid X) = \prod_i \mathbf{P}(E_{Y_i|X} \mid X)$

  $\mathbf{P}(E_X^- \mid X) = \prod_i \sum_{y_i} \sum_{z_i} \mathbf{P}(E_{Y_i|X} \mid X, y_i, z_i)\mathbf{P}(y_i, z_i \mid X)$

- Decompondo $E_{Y_i|X}$ em dois componentes independentes $E_{Y_i}^+$ e $E_{Y_i|X}^-$

  $\mathbf{P}(E_X^- \mid X) = \prod_i \sum_{y_i} \sum_{z_i} \mathbf{P}(E_{Y_i}^- \mid X, y_i, z_i)\mathbf{P}(E_{Y_i|X}^+ \mid X, y_i, z_i)\mathbf{P}(y_i, z_i \mid X)$

  $\mathbf{P}(E_X^- \mid X) = \prod_i \sum_{y_i} \mathbf{P}(E_{Y_i}^- \mid y_i) \sum_{z_i} \mathbf{P}(E_{Y_i|X}^+ \mid z_i)\mathbf{P}(y_i, z_i \mid X)$

Inferência em Redes de Crenças

- Aplicando Bayes a $\mathbf{P}(E^+_{Y_i|X} \mid z_i)$:

  $\mathbf{P}(X \mid E^-_X) = \prod_i \sum_{y_i} \mathbf{P}(E^-_{Y_i} \mid y_i) \sum_{z_i} \frac{P(z_i|E^+_{Y_i|X})P(E^+_{Y_i|X})}{P(z_i)} \mathbf{P}(y_i, z_i \mid X)$

  ...

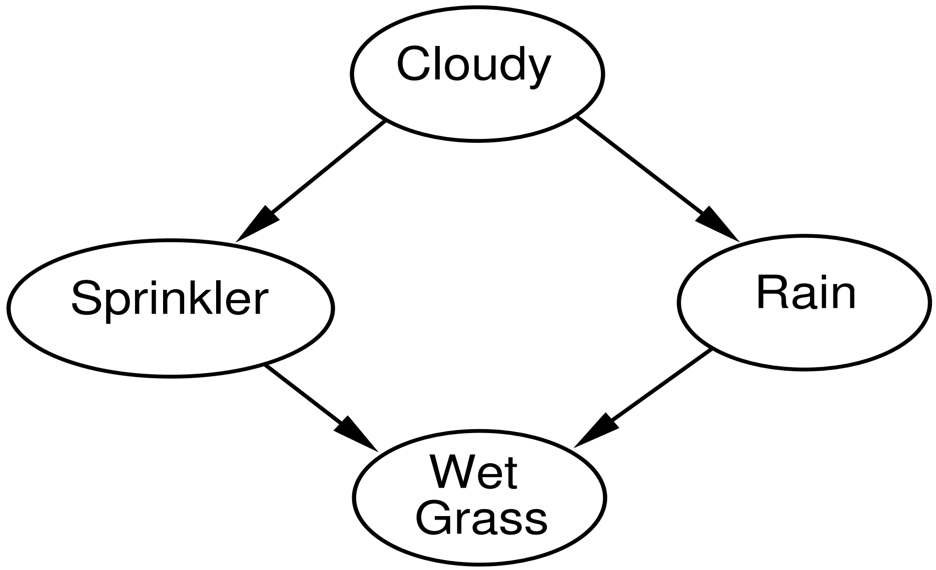
  $\mathbf{P}(X \mid E^-_X) = \beta \prod_i \sum_{y_i} P(E^-_{Y_i} \mid y_i) \sum_{z_i} P(y_i \mid X, z_i) \prod_{z_i} P(z_{ij} \mid E_{Z_{ij}|Y_i})$

- $P(E^-_{Y_i} \mid y_i)$ é uma instância recursiva de $P(E^-_X \mid X)$.
- $P(y_i \mid X, z_i)$ é tirada diretamente da TPC de Y_i.
- $P(z_{ij} \mid E_{Z_{ij}|Y_i})$ é uma instância recursiva de $P(X \mid E)$.

Inferência em Redes de Crenças Multiplamente conectadas

P(C) = .5

C	P(S)
T	.10
F	.50



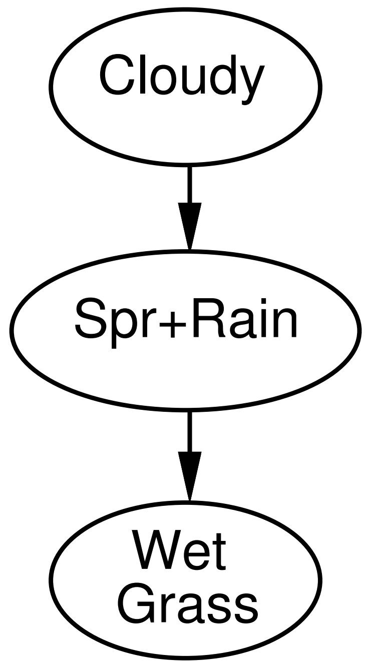
C	P(R)
T	.80
F	.20

S	R	P(W)
T	T	.99
T	F	.90
F	T	.90
F	F	.00

Inferência em Redes de Crenças Multiplamente conectadas

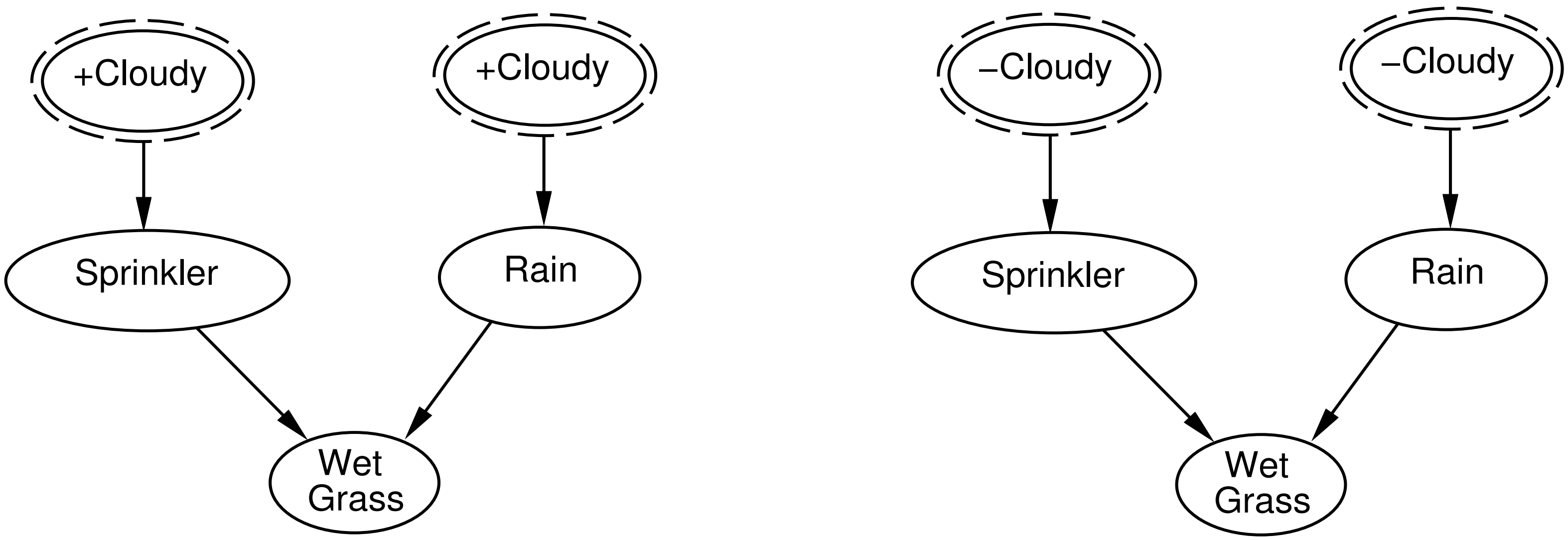
- Três classes de algoritmos:
  - **Clustering**: transforma a rede em uma poli-árvore probabilisticamente equivalente, mas com topologia diferente.
  - **Condicionamento**: oposto de clustering, transforma a rede em várias poli-árvores através da instanciação de valores para as variáveis aleatórias. Avalia a poli-árvore para cada instância diferente.
  - **Simulação estocástica** (ou amostragem lógica): calcula uma prob aproximada através de simulações repetidas do mundo descrito pela rede, observando a frequência com que eventos relevantes acontecem.
  - Em geral: inferência exata em redes de crenças é um problema NP-difícil.

P(C) = .5



C	P(S+R=x) TT	TF	FT	FF
T	.08	.02	.72	.18
F	.40	.10	.40	.10

S+R	P(W)
T T	.99
T F	.90
F T	.90
F F	.00


+Cloudy
+Cloudy
Sprinkler
Rain
Wet Grass
−Cloudy
−Cloudy
Sprinkler
Rain
Wet Grass

Engenharia do Conhecimento para Raciocínio Probabilístico

- Decidir sobre o que falar.
- Decidir sobre o vocabulário de variáveis aleatórias.
- Codificar o conhecimento sobre as relações de dependências entre as variáveis.
- Codificar uma descrição de uma instância específica do problema.
- Colocar consultas ao procedimento de inferência.

Exemplo Prático de Rede Bayesiana [Burnside, AMIA 2005]

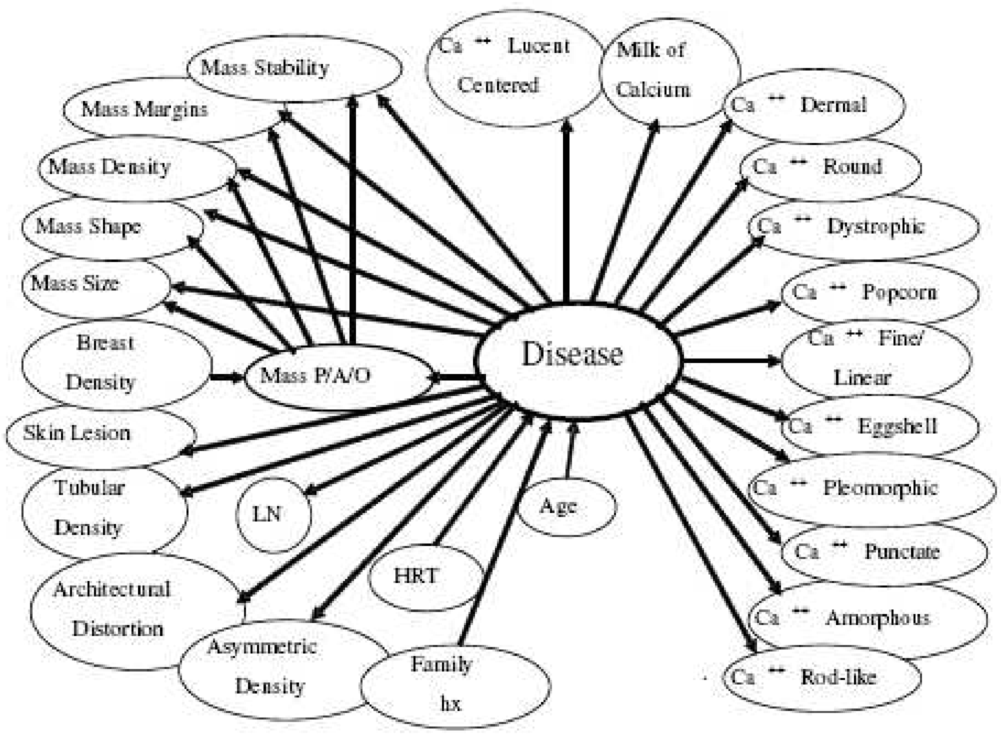


Figure 4.1 Expert Bayesian Network for Mammography

Outras Abordagens

- Por que? Falta de credibilidade em métodos probabilísticos para solução de problemas devido ao crescimento exponencial das tabelas e algoritmos de redes de crenças ainda não conhecidos.
- Probabilidade é numérica (!). Raciocício humano é mais "qualitativo" que "quantitativo". Principal abordagem qualitativa: **default reasoning**, conclusões não são confiáveis com certo grau, mas acreditadas até que uma razão melhor é encontrada para acreditar em outra coisa.
- Sistemas **baseados em regras**: baseados em sistemas lógicos baseados em regras, mas com anotações nos programas para representar incerteza.

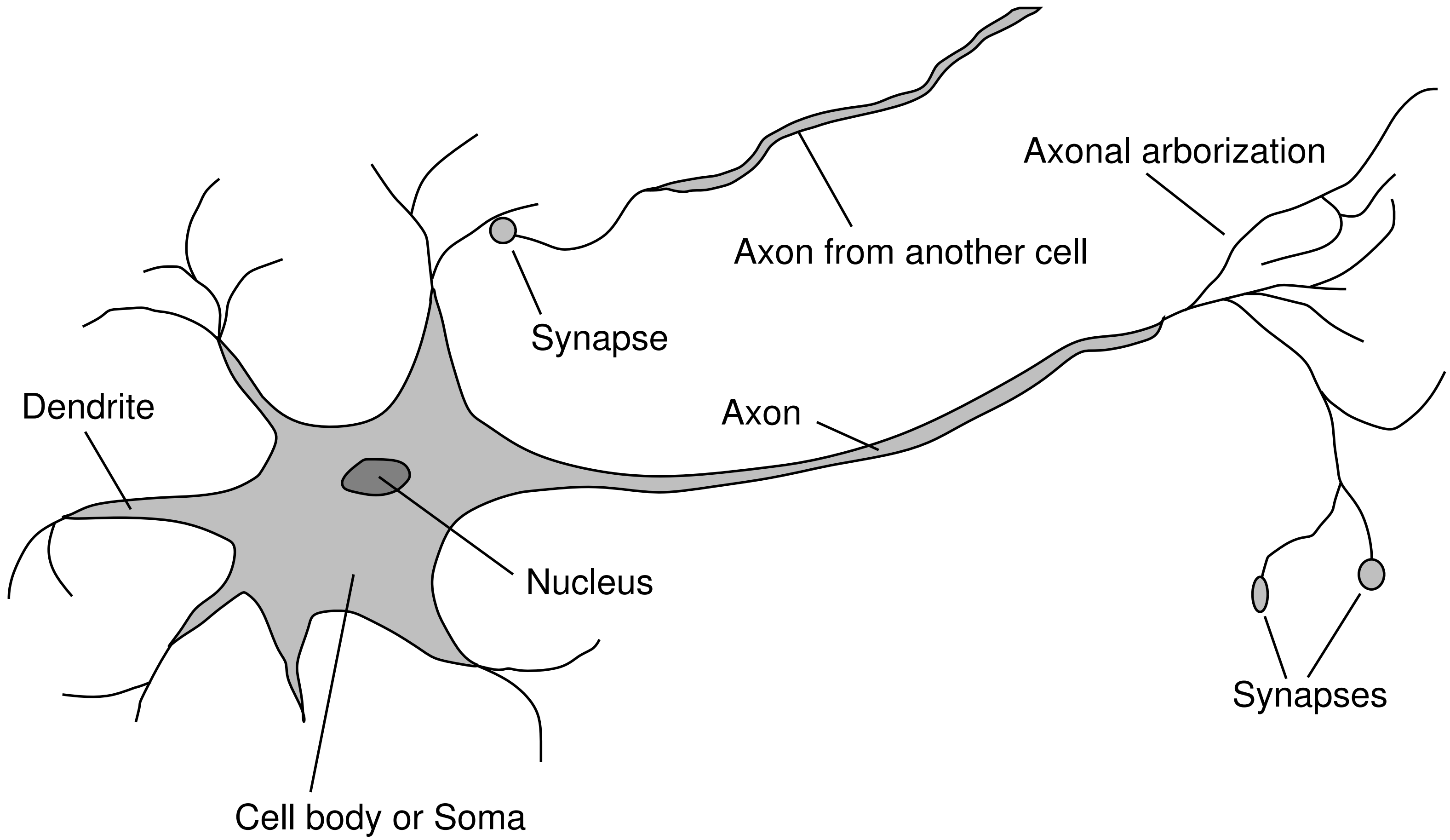
Outras Abordagens

- **Dempster-Shafter theory**: utiliza graus de confiança em intervalos (além de tratar de incerteza, trata de ignorância).
- **Lógica Nebulosa**: representa o conhecimento "vago". Um evento pode ser +/- verdadeiro. Por exemplo, $T(A \wedge B) = min(T(A), T(B))$, onde T é uma função de verdade nebulosa. $T(A \vee \neg A) \neq T(True)$. Ex de linguagem: FRIL (Univ of Bristol, Jim Baldwin, Lotfi Zadeh).

Apoio à Decisão: Redes Neuronais

Aprendendo em Redes Neuronais e Redes de Crenças

- Do ponto de vista computacional: métodos para representar funções usando redes de elementos aritméticos simples, e aprender tais representações através de exemplos.
- Do ponto de vista biológico: Modelo matemático para a operação do cerébro.
- **Neurônios**: Elementos aritméticos simples.
- **Redes Neuronais**: conj de neurônios interligados.


Axonal arborization
Axon from another cell
Synapse
Dendrite
Axon
Nucleus
Synapses
Cell body or Soma

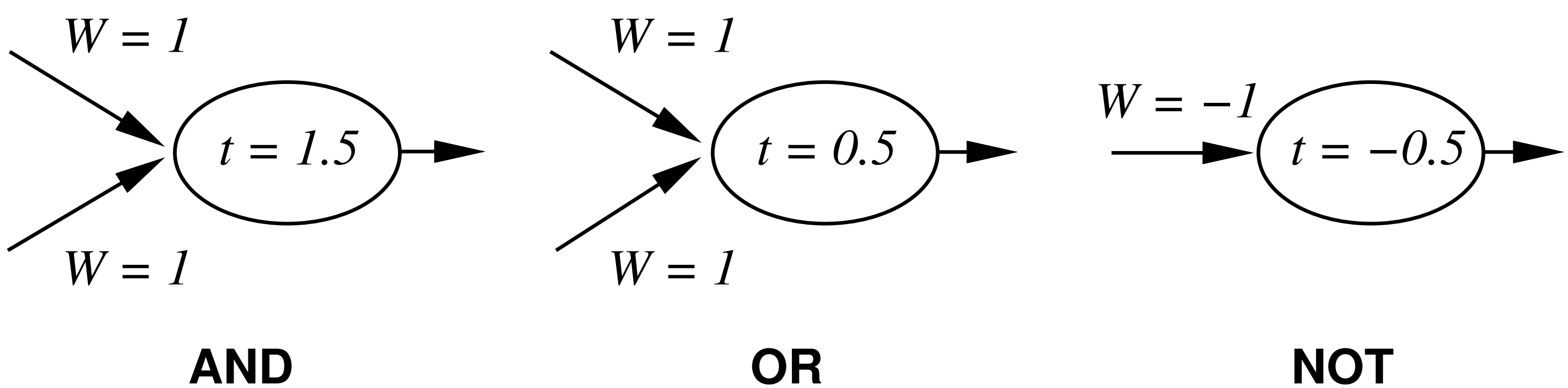
	Computer	Human Brain
Computational units	1 CPU, 10^5 gates	10^{11} neurons
Storage units	10^9 bits RAM, 10^{10} bits disk	10^{11} neurons, 10^{14} synapses
Cycle time	10^{-8} sec	10^{-3} sec
Bandwidth	10^9 bits/sec	10^{14} bits/sec
Neuron updates/sec	10^5	10^{14}

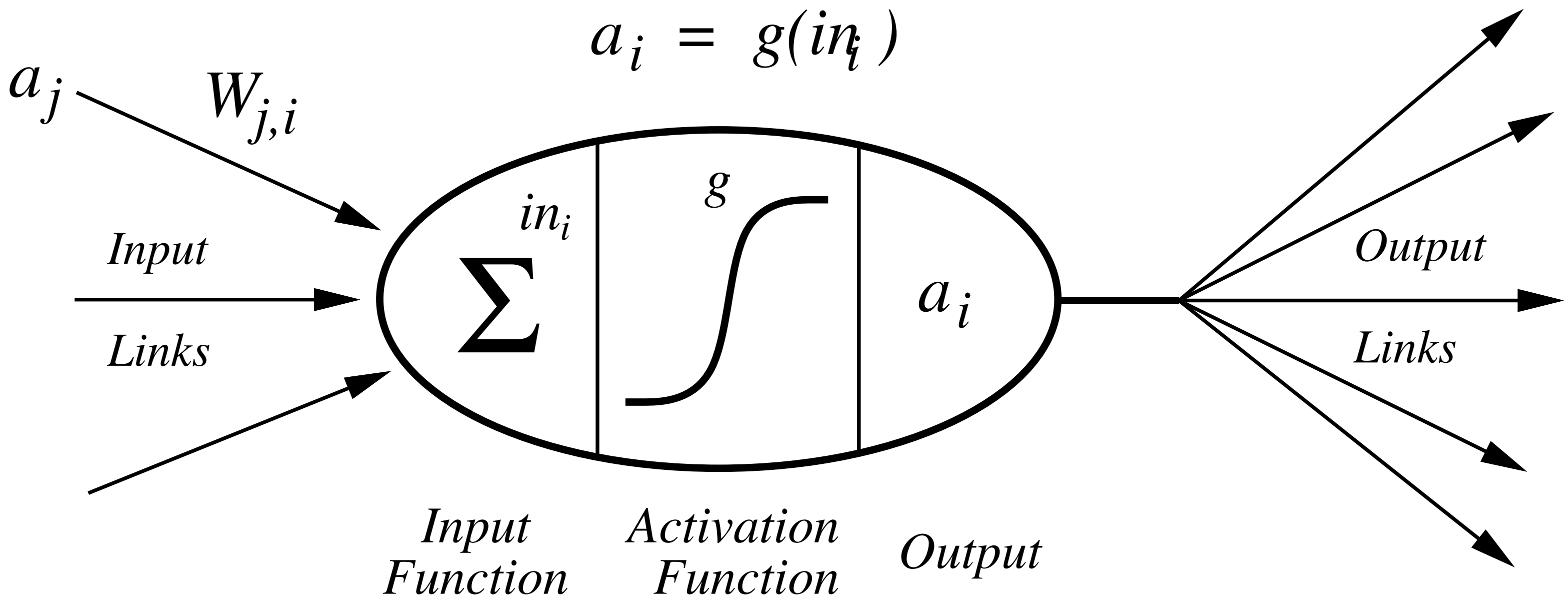
Aprendendo em Redes Neuronais e Redes de Crenças

- Args a favor de redes neuronais:
  - Esperança de que um dispositivo possa ser construído de forma a combinar o paralelismo inerente do cérebro com trocas rápidas de contexto.
  - Redes neuronais podem prover um modelo de paralelismo massivo em contraste com paralelização de algoritmos tradicionais.
  - **Degradação gradual**: se alguma coisa dá errada, o desempenho decresce gradualmente.
  - Projetadas para serem treinadas utilizando algoritmos de aprendizado indutivo.

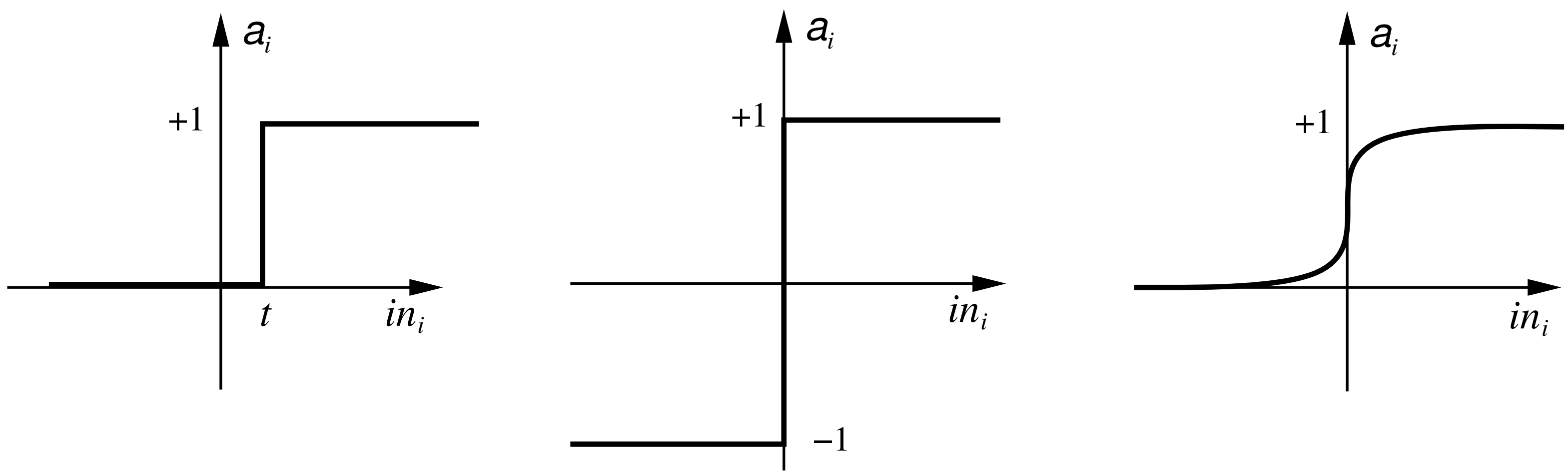
Redes Neuronais

- Para construir uma rede: número de unidades, tipos de unidades, formato das conexões.
- Próximo passo: inicializar os pesos da rede, e treinar os pesos usando um algoritmo de aprendizado aplicado a um conj de treinamento para a determinada tarefa implementada pela rede.
- A operação de unidades individuais pode ser comparada com portas lógicas (McCulloch and Pitts).





$a_i = g(in_i)$
a_j
$W_{j,i}$
Input
Links
in_i
Σ
g
a_i
Output
Links
Input
Function
Activation
Function
Output



(a) Step function **(b) Sign function** **(c) Sigmoid function**