x <- x_0 + alpha * gradient of f(x_0)
independence P(A|B) - P(A)
cond indep P(A,B|C) = P(A|C)P(B|C)
bayes P(A|B) = P(B|A)P(A)/P(B)
defn P(A|B) = P(A,B)/P(B)

PEAS - 41 (3.1)
Performance, Environment, Actuators, Sensors

Environment Types - 42-48 (3.2)
- Fully (lab 1) vs Partially (lab 2) observable
- deterministic (next state based only on current state and next action)/stochastic
- uncertainty - partially observable or non-deterministic
- Static (world doesn't change)/dynamic
- semidynamic - environment is same but time affects performance (like chess with timer)
- discrete/continuous
- single/multiple agents - communication, competitive/cooperative, randomization

CSA -
States
Actions
Consequences
Goals
Preferences - map consequences to utility
Utility - point system

Agent Types 49-58 (4.2)
- (A) simple reflex - condition action rules (if then) - ignores history (vacuum cleaner, p controller)
- (SAC) model based reflex agent - same as previous but has history/state (vacuum cleaner)
- (SACG) goal based - info about goal, searching & planning (potential field tanks, pd controller, searches (at least goal based, maybe more),)
- (SACGU) utility based - happiness level
- learning agent - has ^^ model in it, learns if it's getting better or worse, then changes

rationality & optimization - 38 (pre 3.1)
1) performance measure is criterion for success
2) agent has prior knowledge
3) actions that agent can perform
4) agents precept sequence from past

PID controller
Potential - current error
Differential - last error
Integral - previous errors

Potential Fields - d = distance to goal/obs, θ = arctan(y_g-y / x_g-x),
- attractive - if d<r x=y=0; if r<=d<=s+r x=alpha(d-r)cosθ and y=alpha(d-r)sinθ; if d>s+r x=alpha*s*cosθ y=alpha*s*sinθ
- repulsive - if d<r x=-sign(cosθ)inf y=-sign(sinθ)inf; if r<=d<=s+r x=-Beta(s+r-d)cosθ y = -Beta(s+r-d)sinθ if d > s + r x=y=0
- tangential - same as repulsive with sin/cos swapped
- uniform - constant to everything
- random - randomly generate small field

search - 83 (4.1) - informed searches have a heuristic function; g(x) = cost to arrive, h(x) = cost to goal = heuristic
- Breadth First - uninformed - complete, optimal assuming equal cost - queue
- Depth First - uninformed - complete if finite graph or tree with loop checks , not optimal - stack
- uniform cost search - uninformed - BFS/Dijkstra - goes down shortest current path - complete & optimal
- DFS limited - uninformed - arbitrary depth limit - same complete/optimal as DFS
- Iterative deepening - uninformed - start at 0 depth, increment depth until goal is found - same complete/optimal as DFS
- greedy - informed - incomplete, non optimal
- A* - h(x) + g(x) - g(x) = cost to reach node - complete - optimal
- IDA* - f(g+h) as cutoff for iterative deepening (google maps) - complete and optimal assuming iteration continues until you get there
- Recursive best first search - keeps track of best alternate - optimal, complete (lots of thrashing potential)
- SMA* - simplified memory bounded A* - optimal and complete if solution fits in memory

Heuristic functions:
- admissible - optimistic - never overestimate
- consistent - monotonicity - triangle equality - h(x) <= c(x,a,x') + h(x')

Genetic algorithms:
- path length - number of directions in the gene
- number of genes - number of genes considered
- max generations - how many generations the code runs

Joint probability
P(A,B,C)
Marginal probability
P(A,B)
sum(P(A,B,C)) for all values of C
conditional probability
P(A|B,C) = P(A,B,C)/P(B,C) = P(A,B,C)/sum(P(A,B,C)) for all values of A
P(A,B|C) = P(A,B,C)/P(C) = P(A,B,C)/sum(sum(P(A,B,C)) for all values of A) for all values of B

P(A|B) = P(A,B)/P(B) = P(B|A)P(A)/P(B)
def of ^^conditional probability ^^Bayes rule

1. from conditional to joint
P(A,B,C) ?

2. margin
P(A|C) ?

3. negation
P(!A|B,C) <= 1-P(A|B,C)

13.3a: If P(a|b,c) = P(b|a,c), then P(a|c) = P(b|c)
P(a|b,c) = P(b|a,c)
P(a,b,c) / P(b,c) = P(a,b,c) / P(a,c)
P(a,c) = P(b,c)
P(a|c) = P(b|c)

conditional independence
P(B,C|A) = P(B|A)P(C|A)
P(B|C,A) = P(B|A)