USC Viterbi
School of Engineering

Acting Under Uncertainty

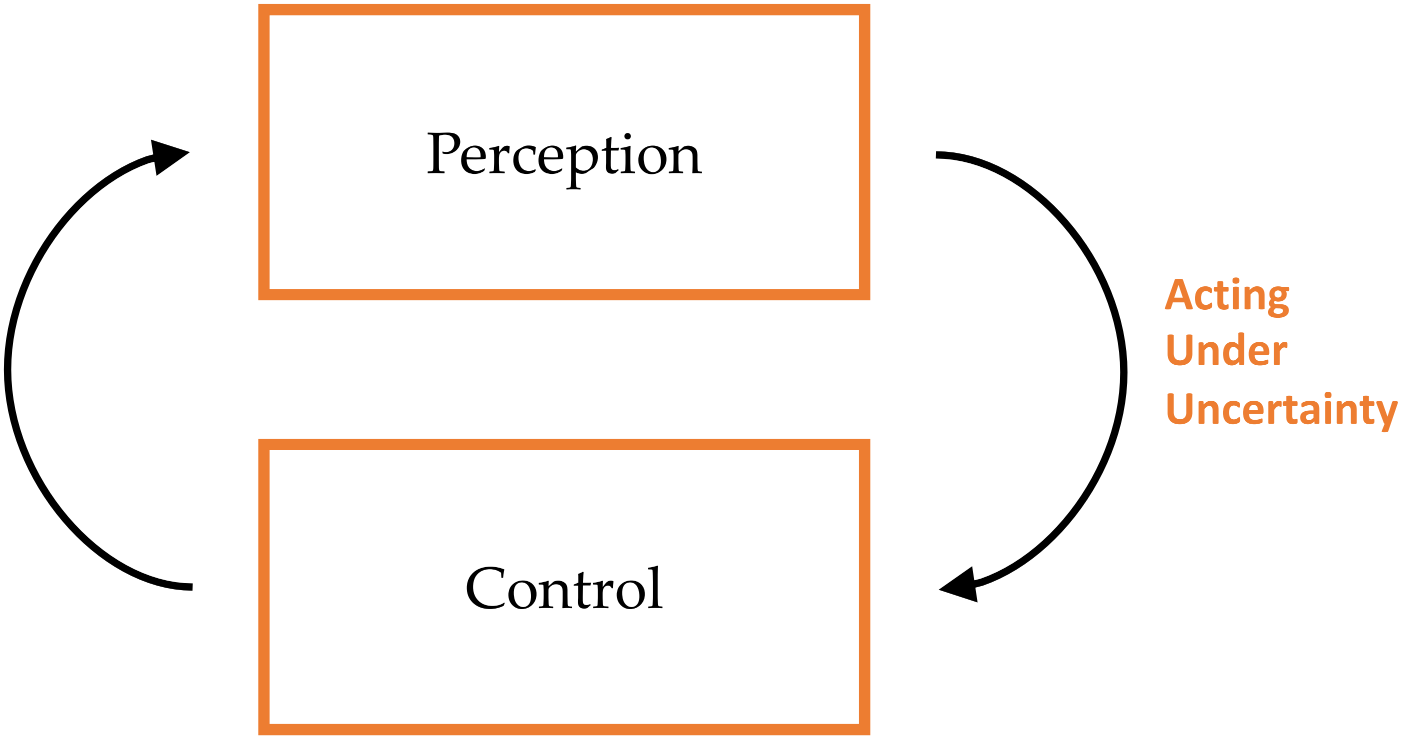
CSCI 545 Introduction to Robotics
Instructor: Stefanos Nikolaidis

[Resources: 6.825 Techniques in Artificial Intelligence, MIT OCW]

USC Viterbi School of Engineering

Bayesian Filtering **Particle Filtering** **Mapping**

KF / EKF **Localization**



Inverse Kinematics **Motion Planning** **Linear / Non-Linear Control**

Task Space Regions **Dynamics**

Uncertainty

USC Viterbi School of Engineering

Causes for uncertainty:

- action outcomes
- state observability

Utility theory provides a principled way to decide on outcomes based on:

- preferences
- belief (probabilities) on states / outcomes

Question

- Would you prefer:
- A sure gain of $240
- A 25% of winning 1000$ and 75% of winning nothing

Question

- Would you prefer:

- A sure gain of $240
- A 25% of winning 1000$ and 75% of winning nothing

$$U(A) = U(240)$$

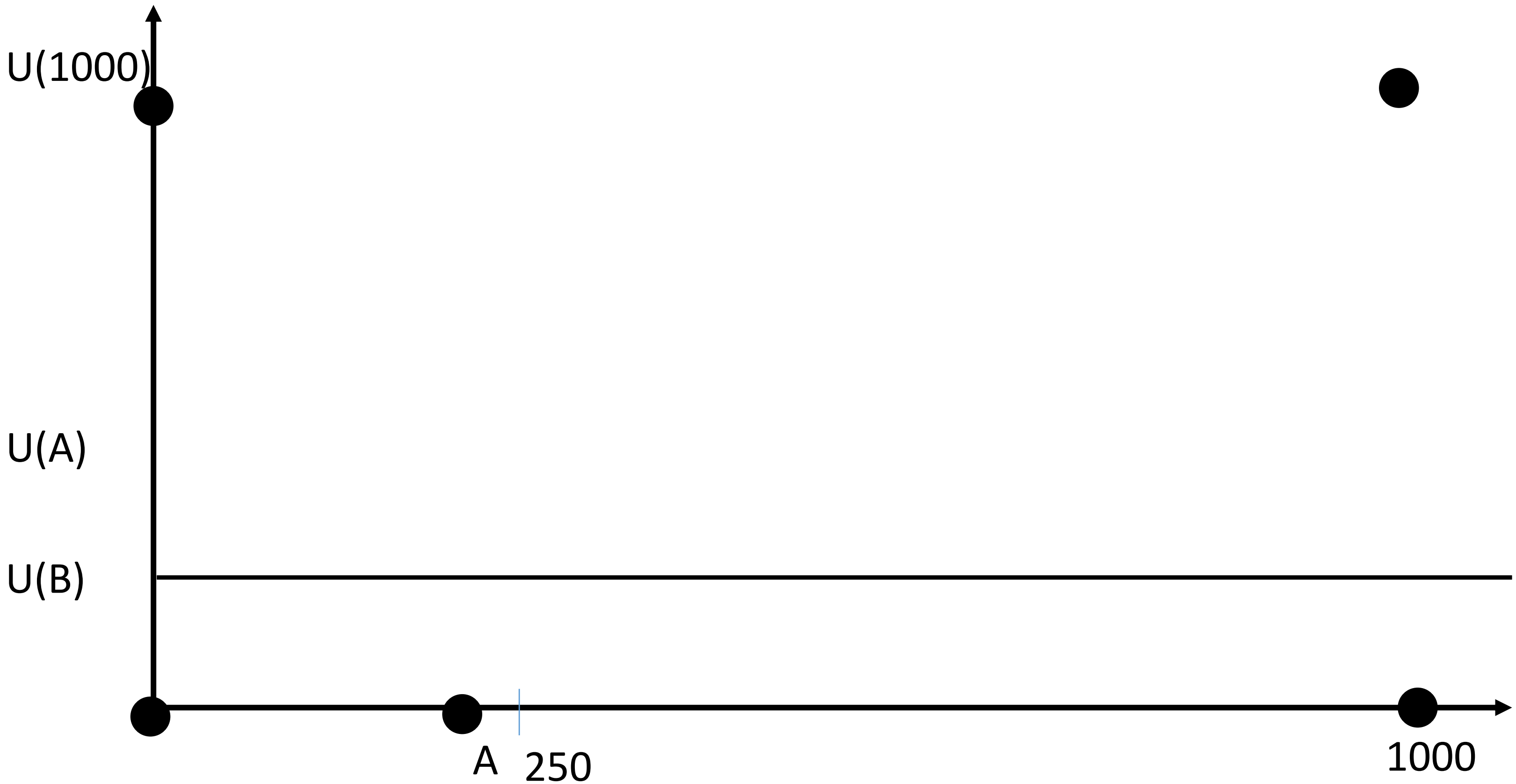
$$U(B) = U(1000) * 0.25 + U(0) * 0.75$$

Utility Theory

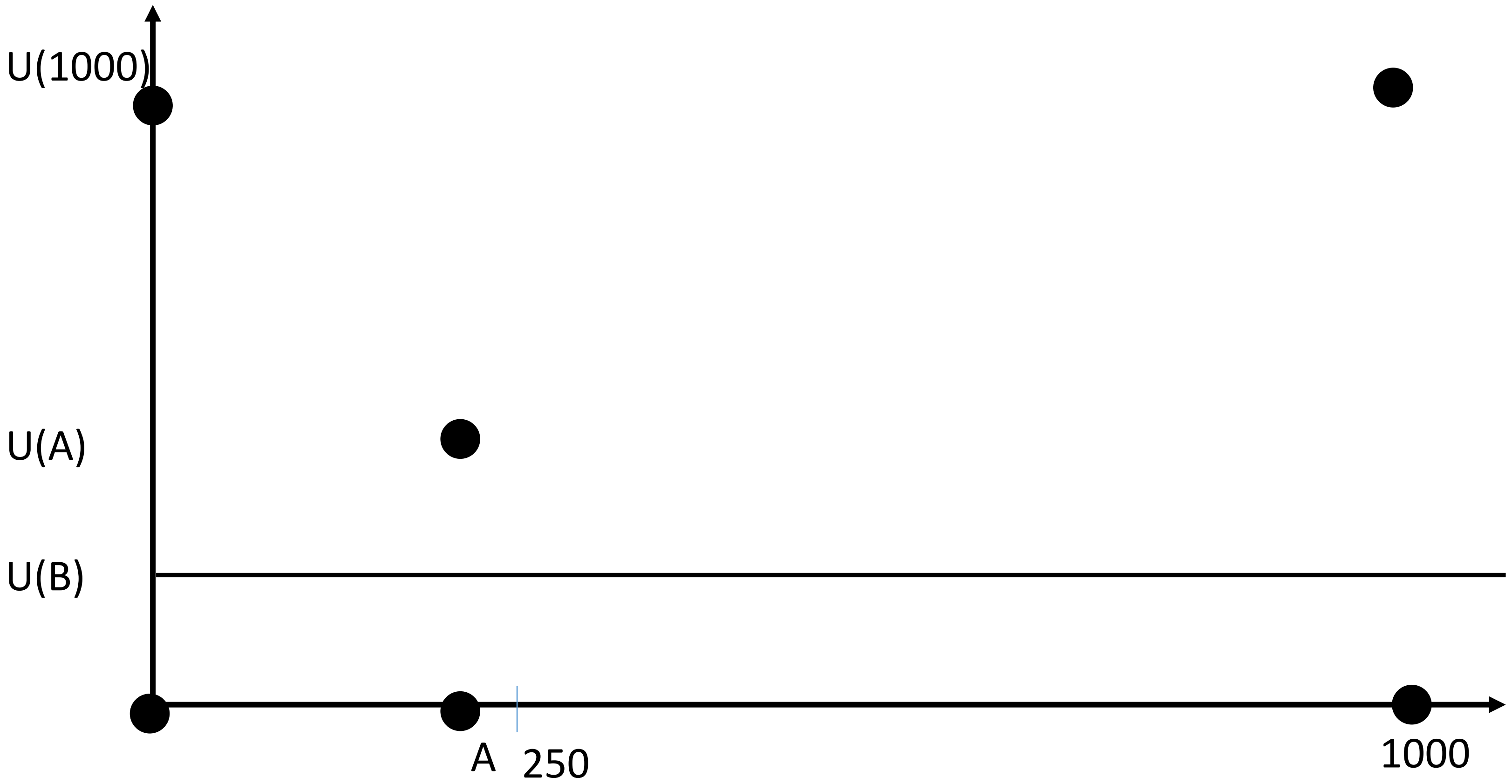
Utility Theory

Utility Theory

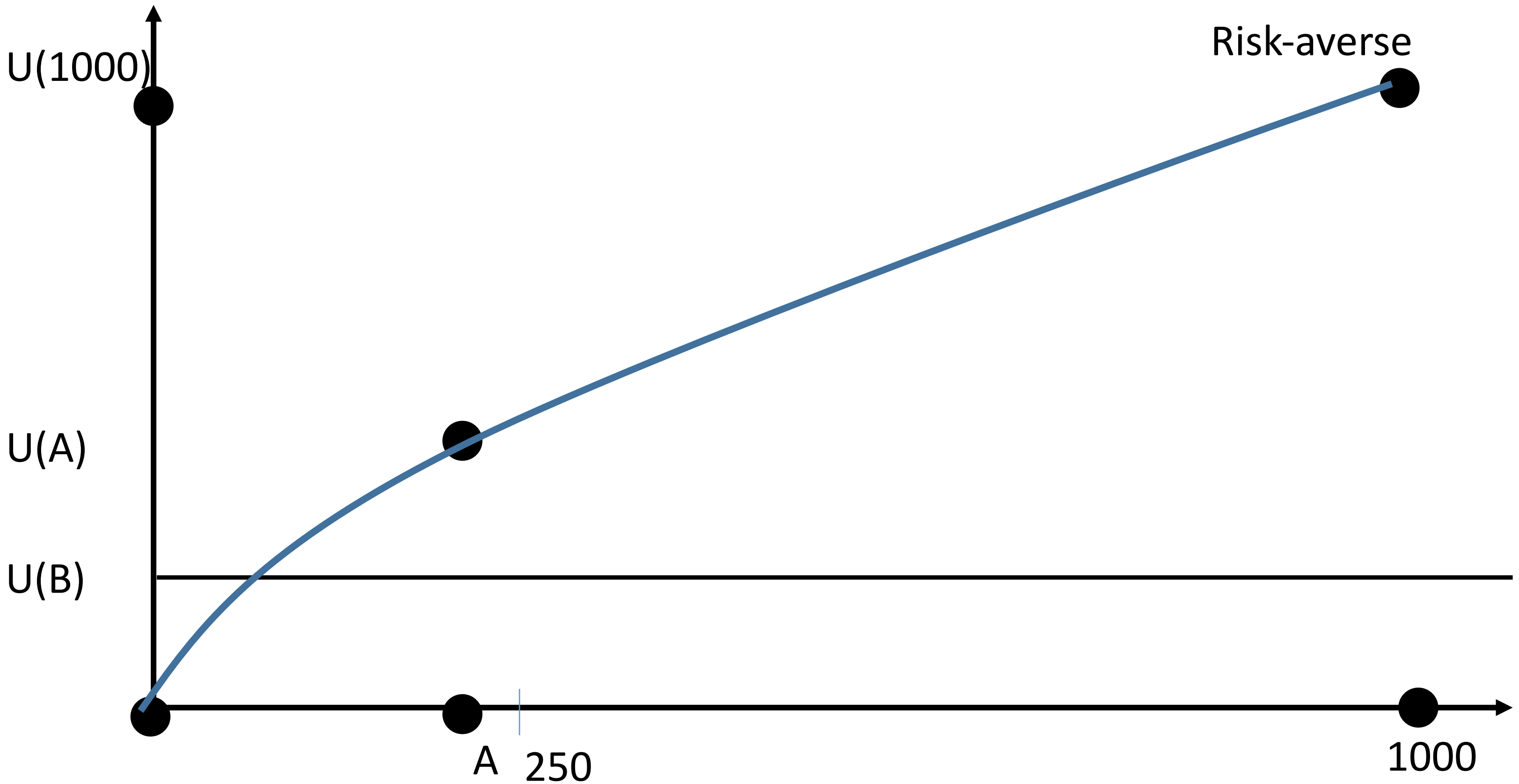
Utility Theory



Utility Theory



Utility Theory



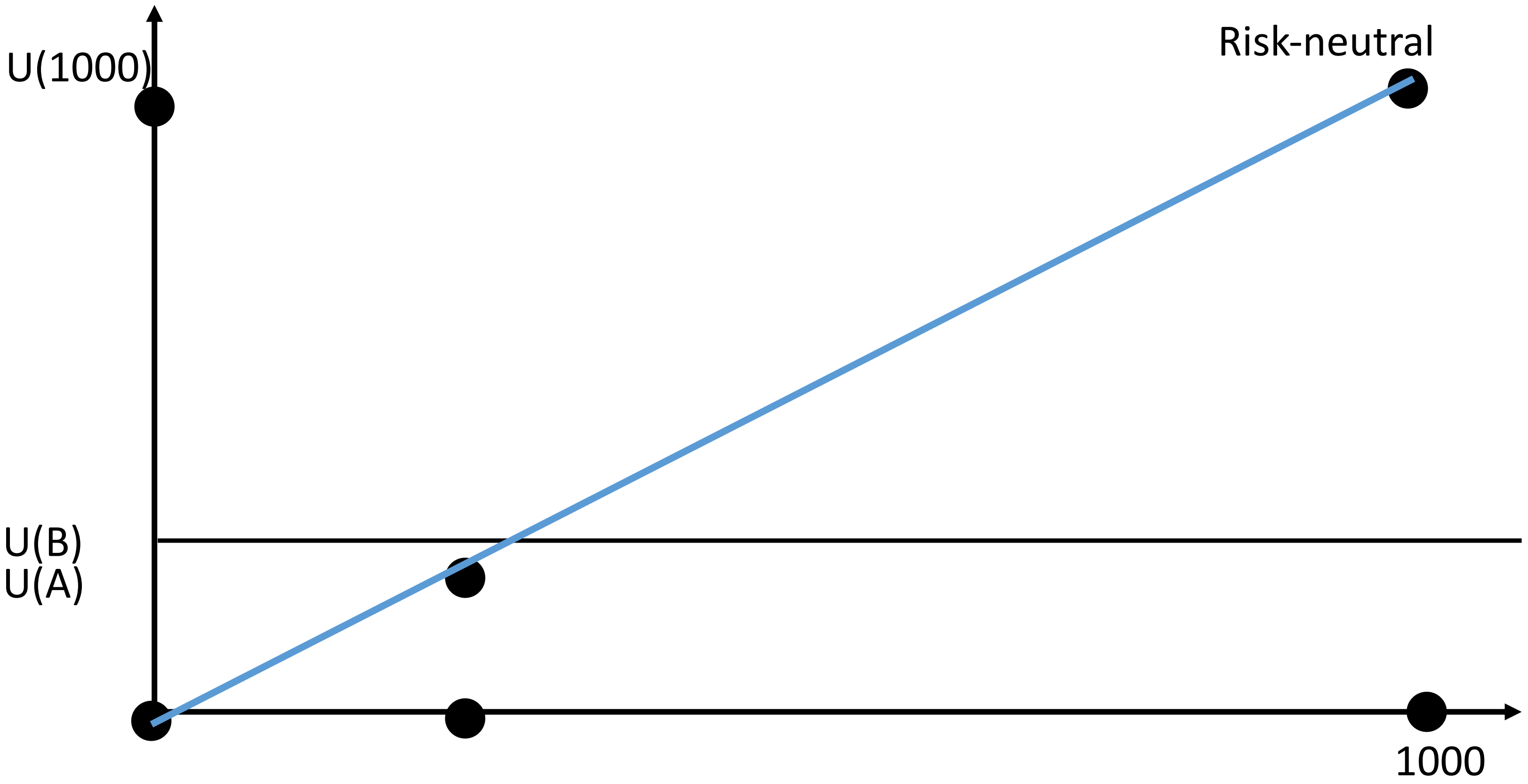
Linear Utility Function

- Would you prefer:
- A sure gain of $240
- A 25% of winning 1000$ and 75% of winning nothing

$$U(A) = 240$$

$$U(B) = 1000 * 0.25 + 0 * 0.75$$

Utility Theory



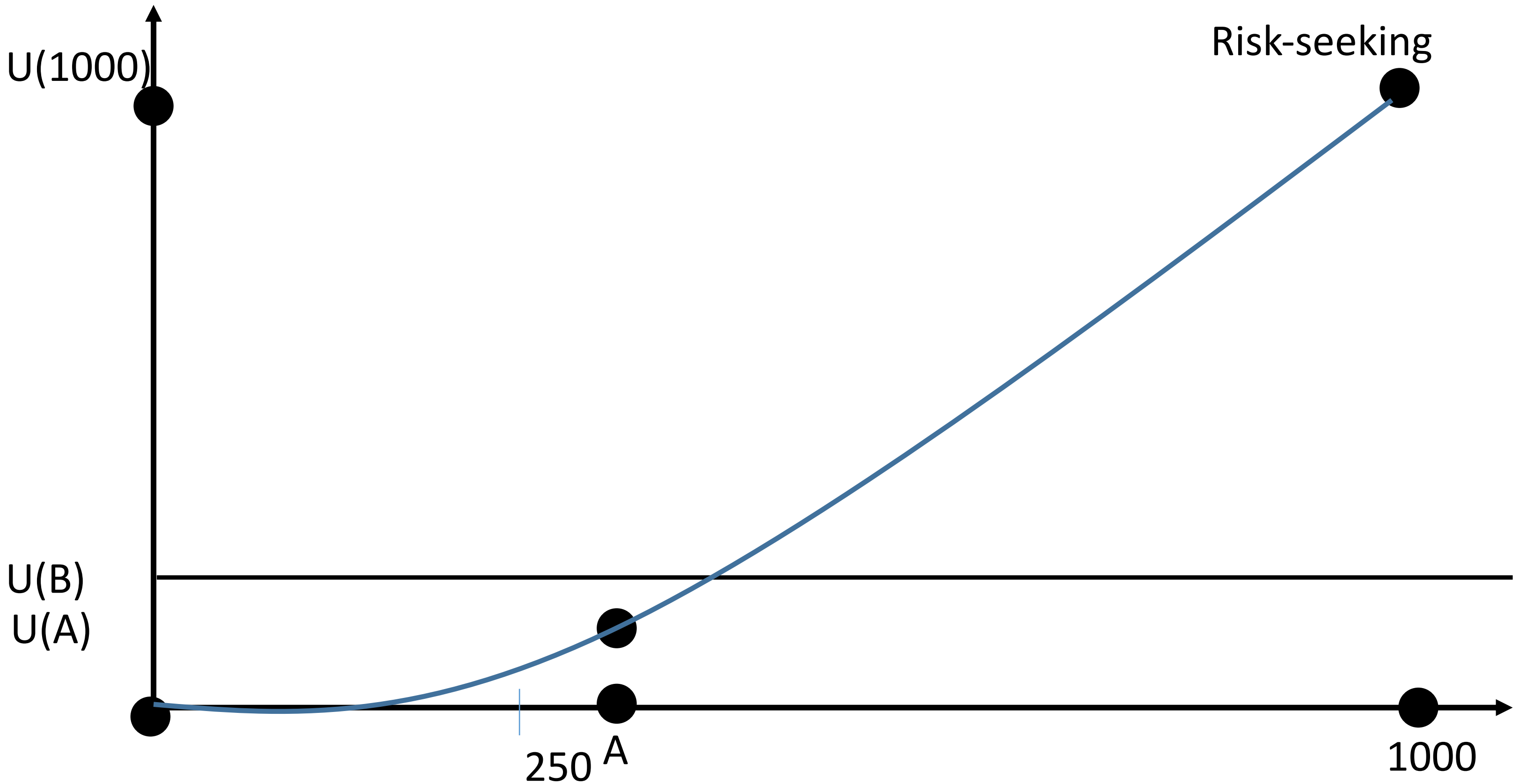
Lottery

- Would you play the lottery:
- A sure gain of $260
- A 25% of winning 1000$ and 75% of winning nothing

$$U(A) = U(260)$$

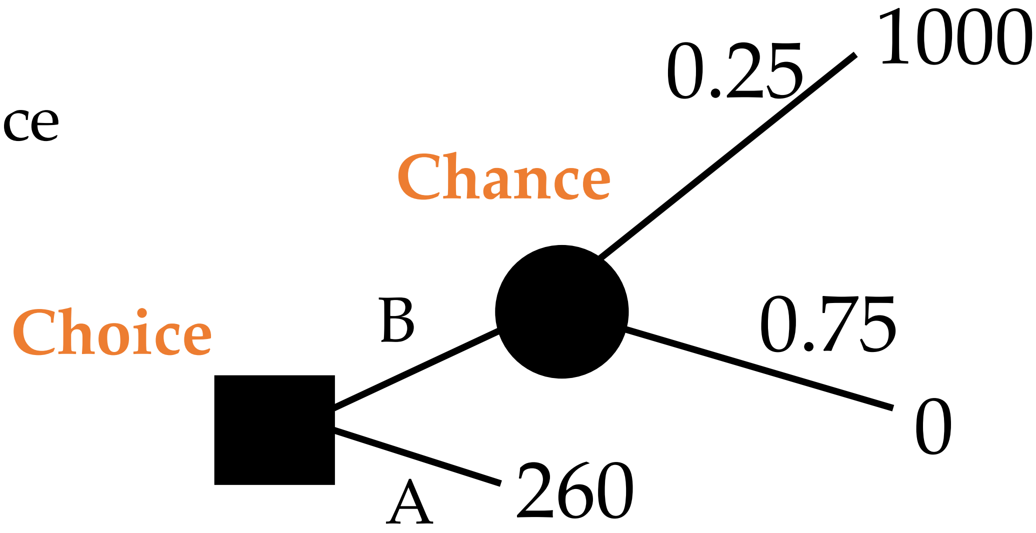
$$U(B) = U(1000) * 0.25 + U(0) * 0.75$$

Utility Theory



Decision Tree

- Stochastic Outcome:
- Choice



Question

- Would you prefer:

A: a sure gain of $240

B: a 25% of winning 1000$ and 75% of winning nothing

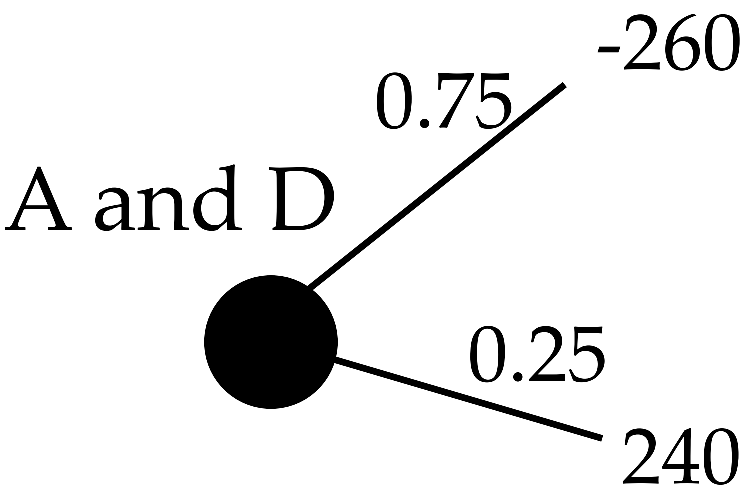
Question 2

- Would you prefer:

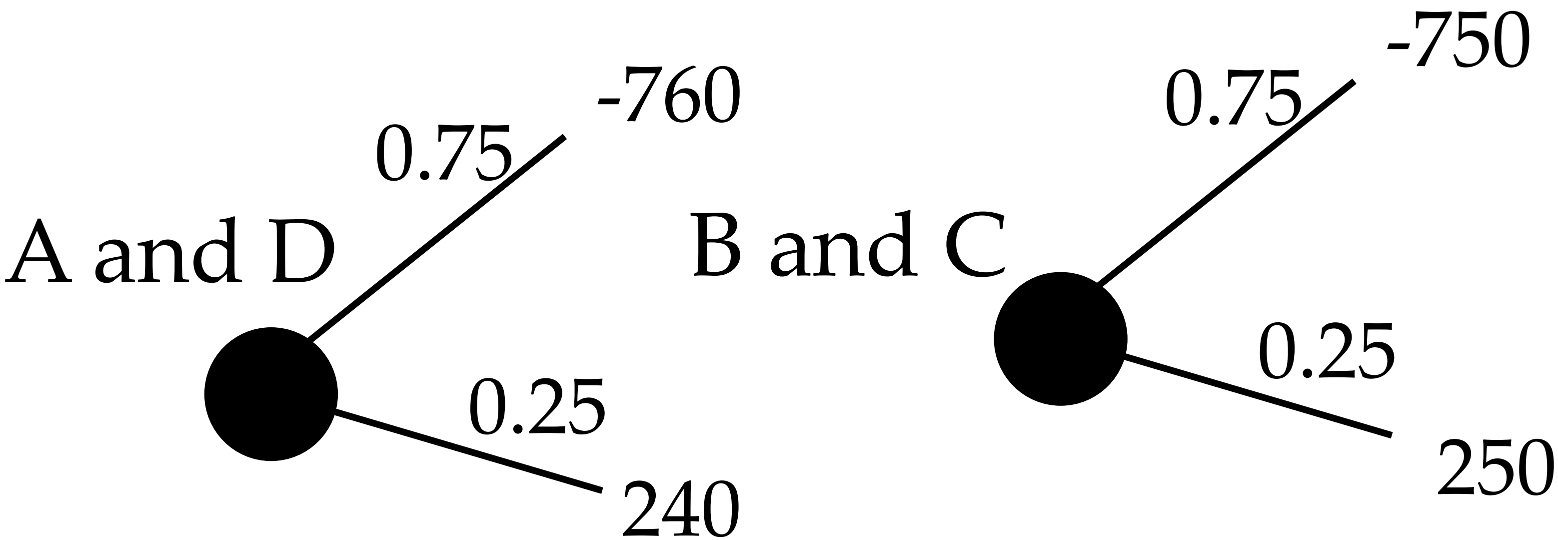
C: A sure loss of $750

D: A 75% of losing 1000$ and 25% of losing nothing

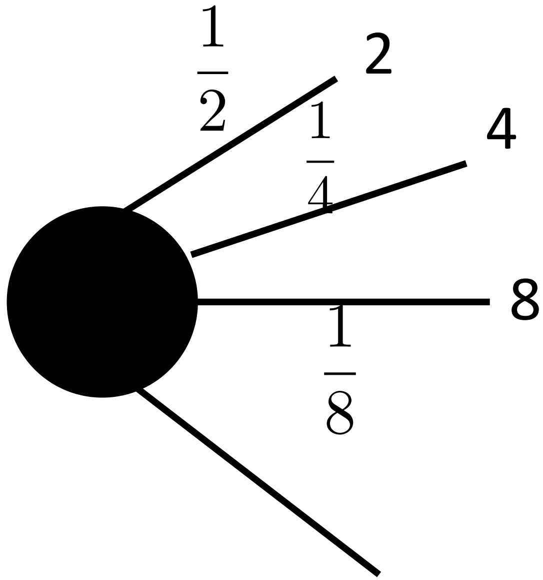
Questions 1 and 2



Questions 1 and 2

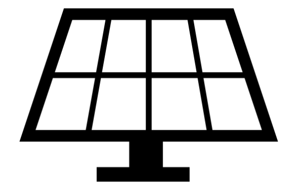


St Petersburgh Paradox

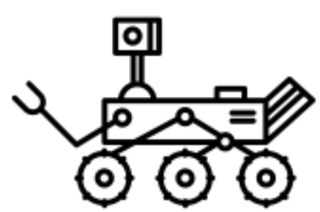


Example

Gains 1100 units from recharging

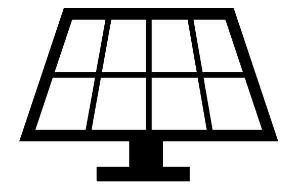


Costs 1000 units to reach a solar panel



Example

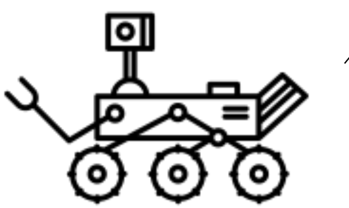
Gains 1100 units from recharging



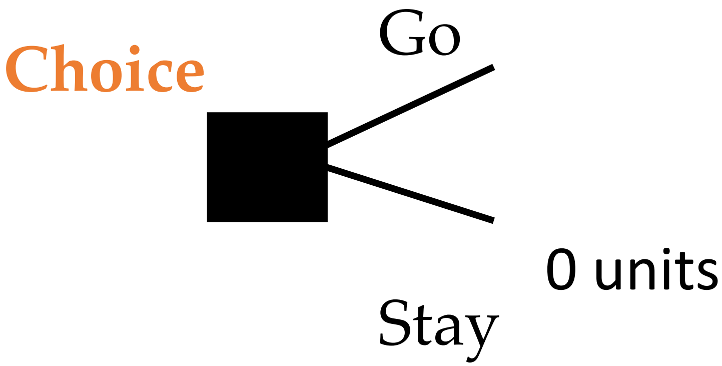
0.8 good panel: minor repair needed (40 units)

0.2 broken panel: major repair needed (200 units)

Costs 1000 units to reach a solar panel

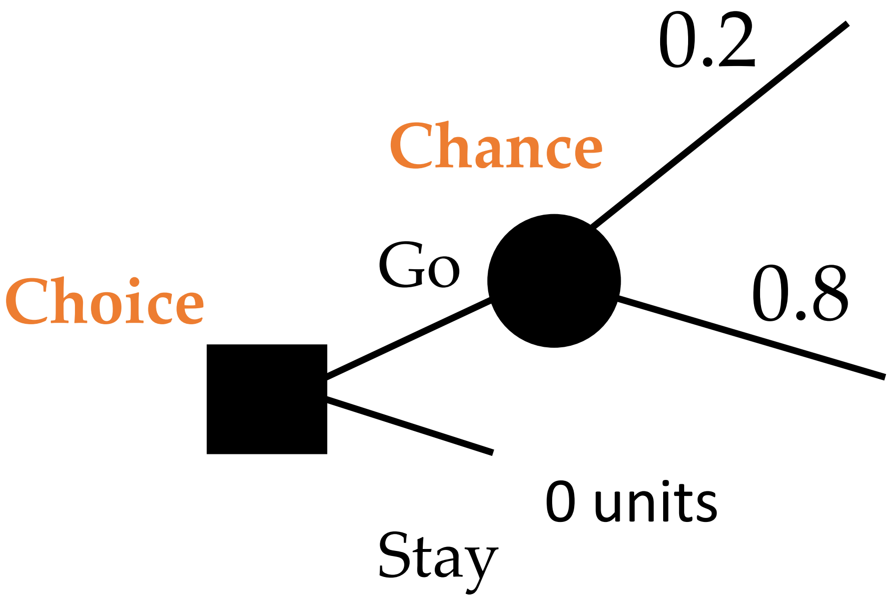


Decision Tree



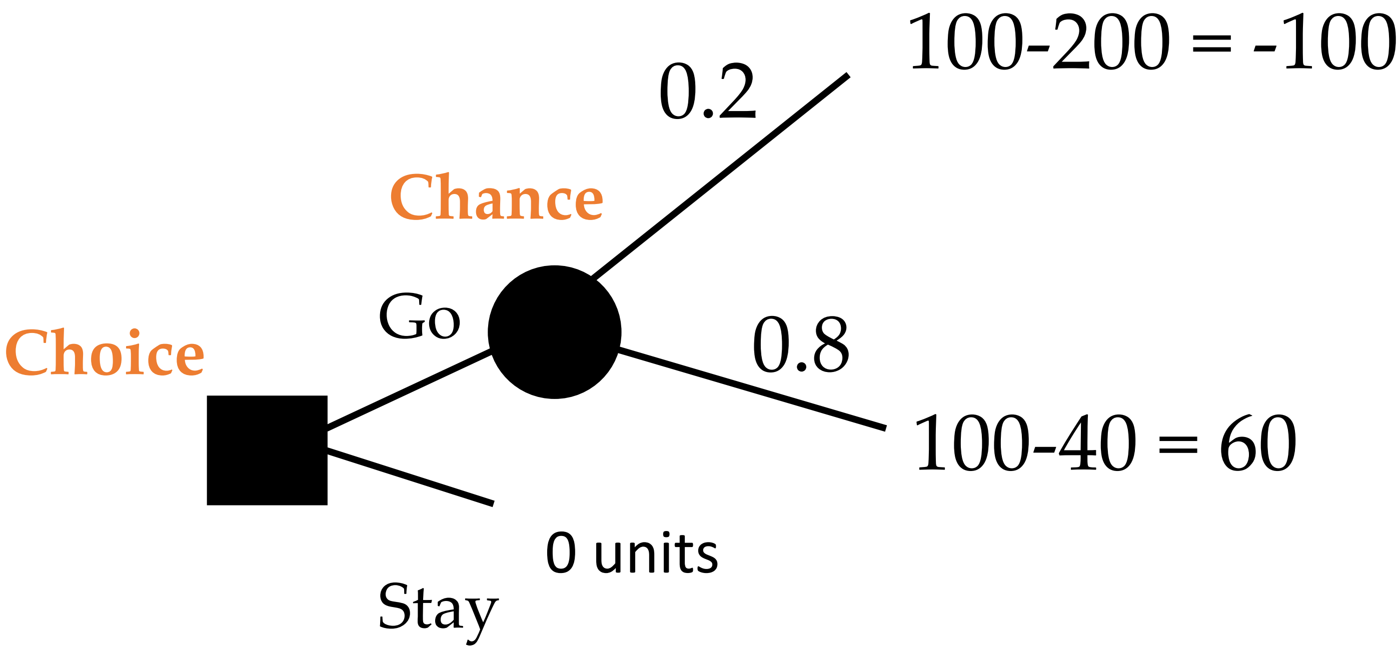
Chance: Expected Value

Decision Tree



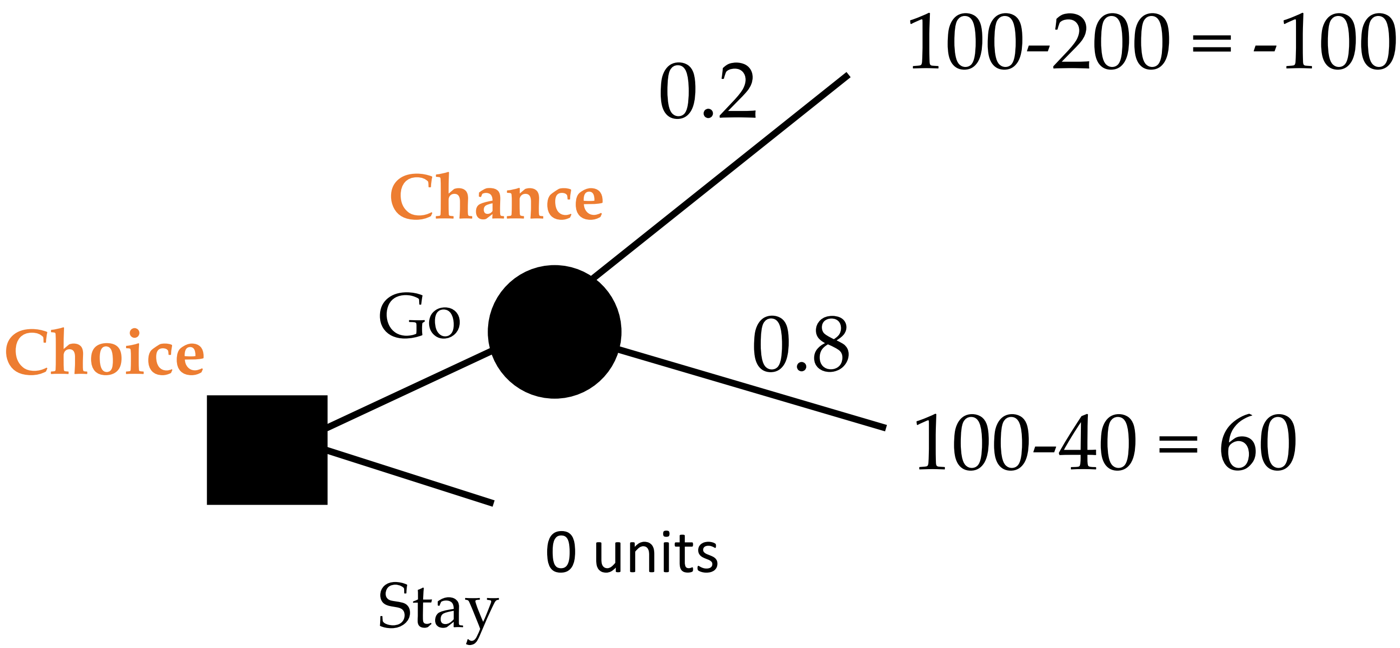
Chance: Expected Value

Decision Tree



Chance: Expected Value

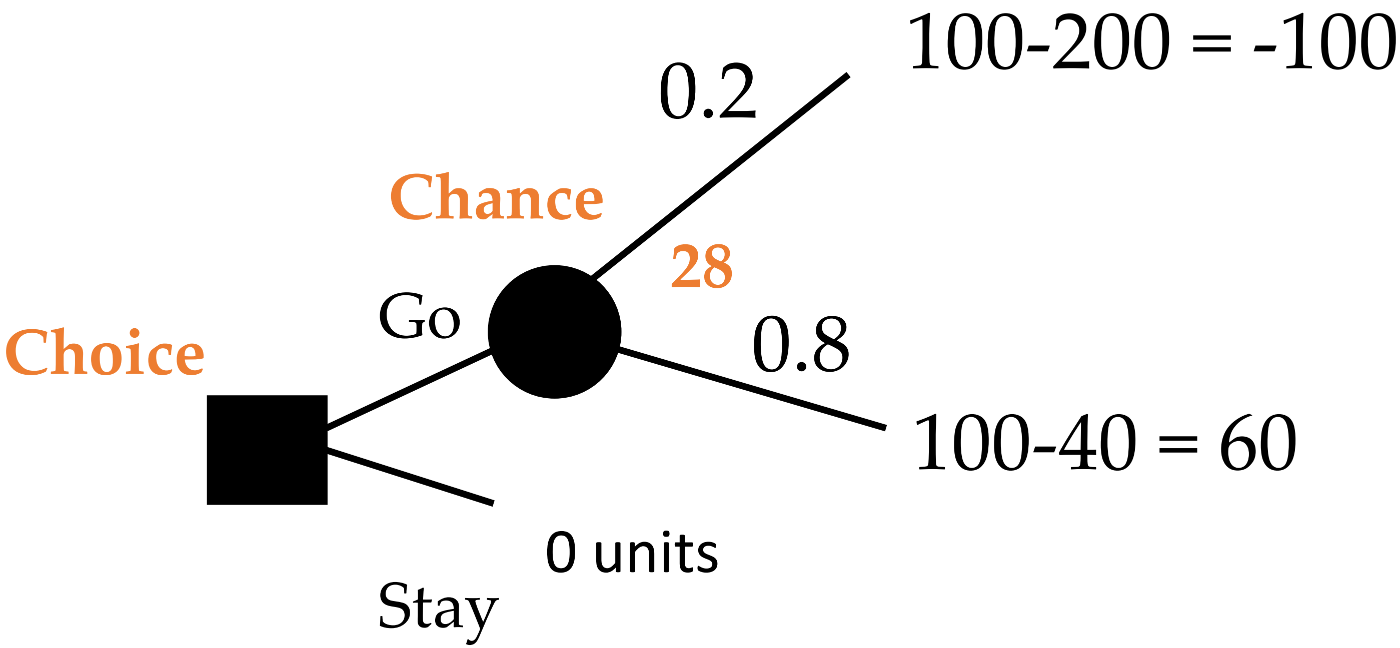
Decision Tree



$$0.2 * (-100) + 0.8 * 60 = -20 + 48 = 28$$

Chance: Expected Value

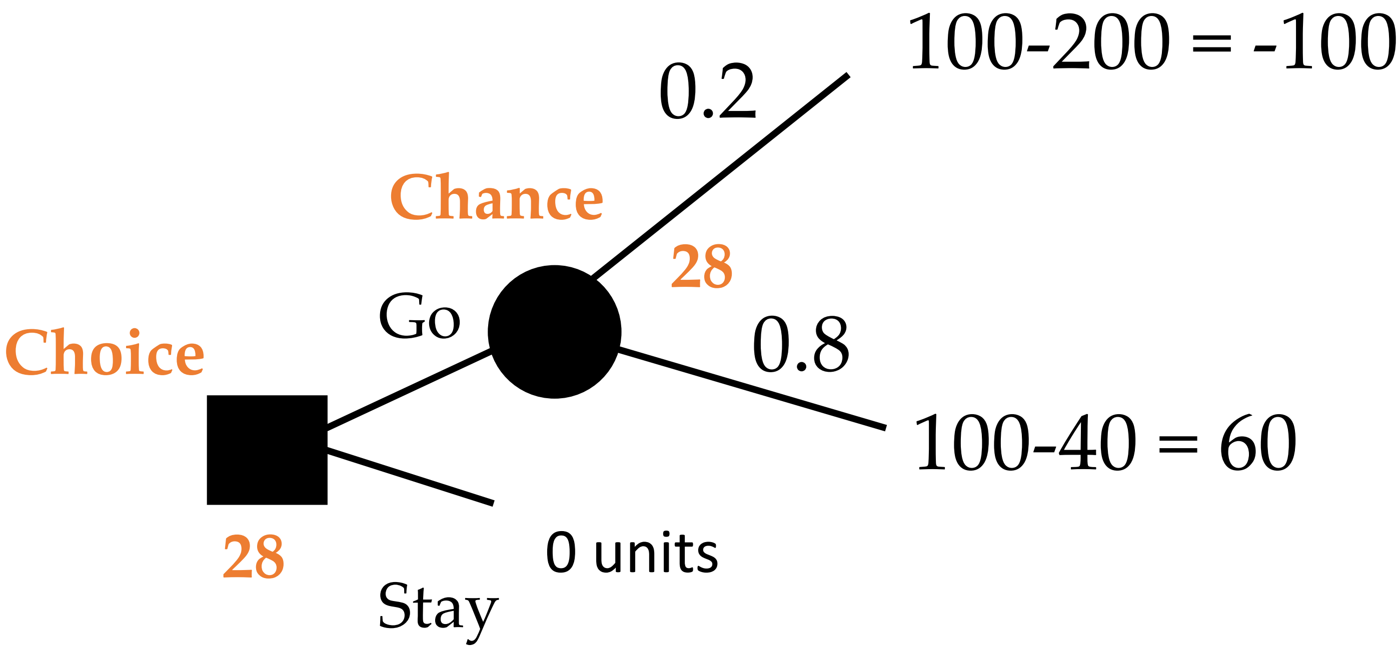
Decision Tree



$$0.2 * (-100) + 0.8 * 60 = -20 + 48 = 28$$

Chance: Expected Value

Decision Tree

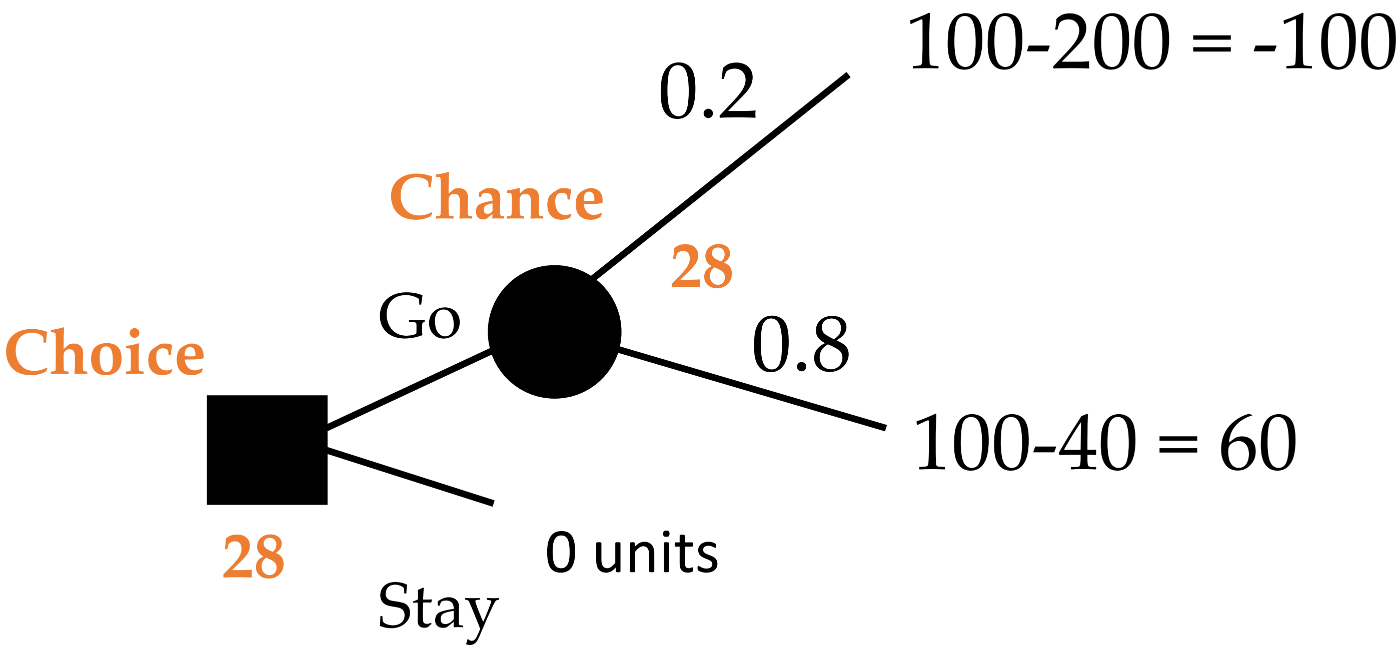


$$0.2 * (-100) + 0.8 * 60 = -20 + 48 = 28$$

Chance: Expected Value

Choice: Maximum Value

Decision Tree

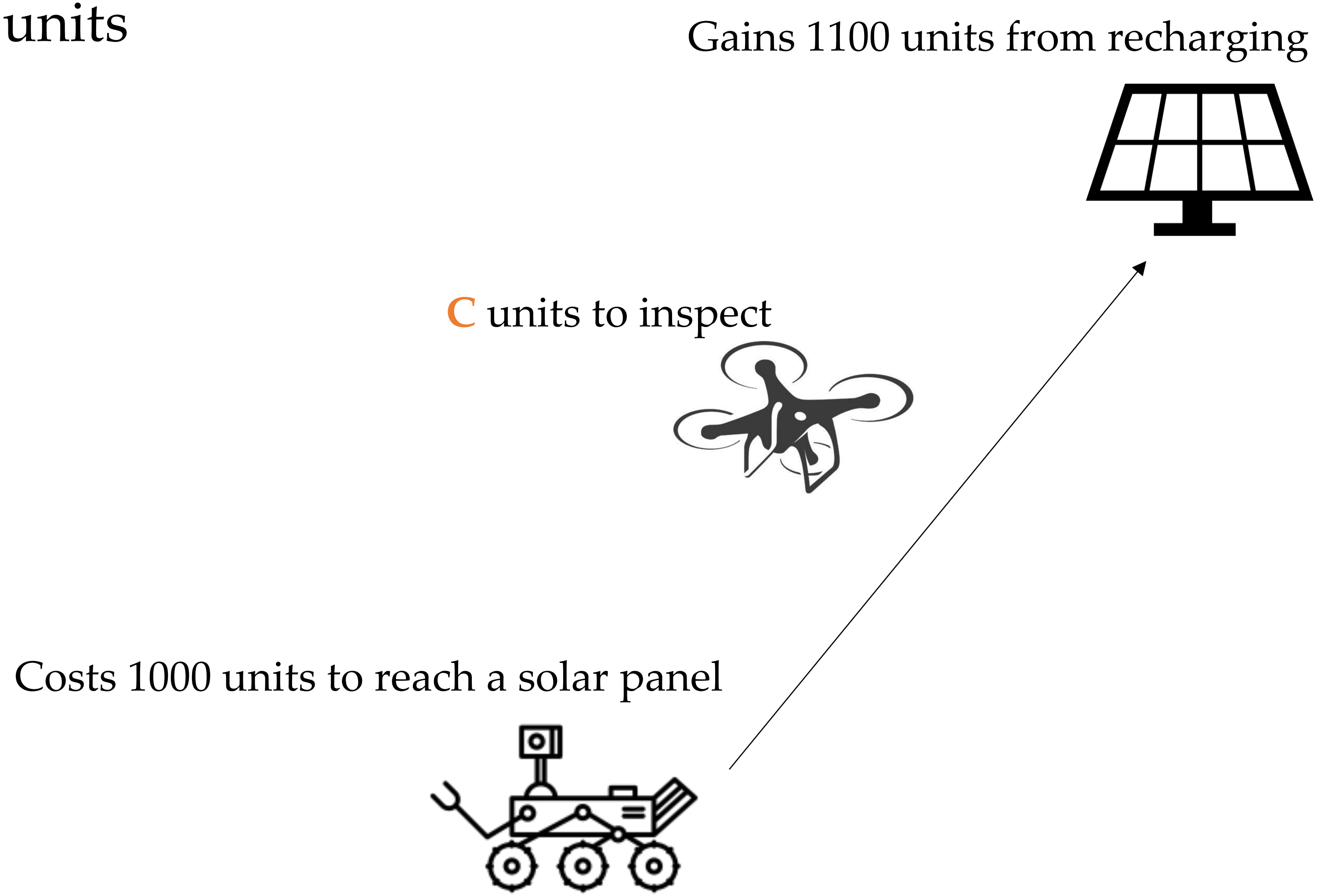


$$0.2 * (-100) + 0.8 * 60 = -20 + 48 = 28$$

What if you are risk-averse?

Expected value of perfect information

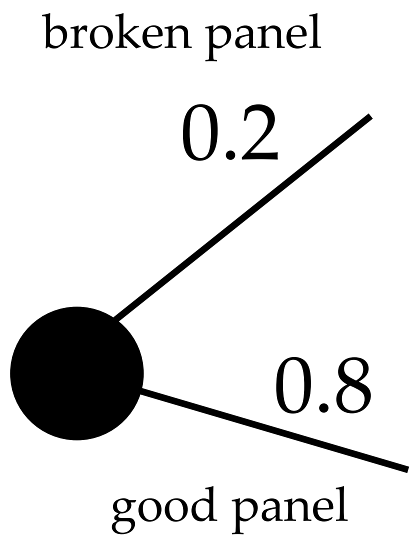
What if the rover can deploy a drone to inspect that will cost C units



Expected Value of Perfect Information

How much would the agent pay for information of what type of solar panel it is?

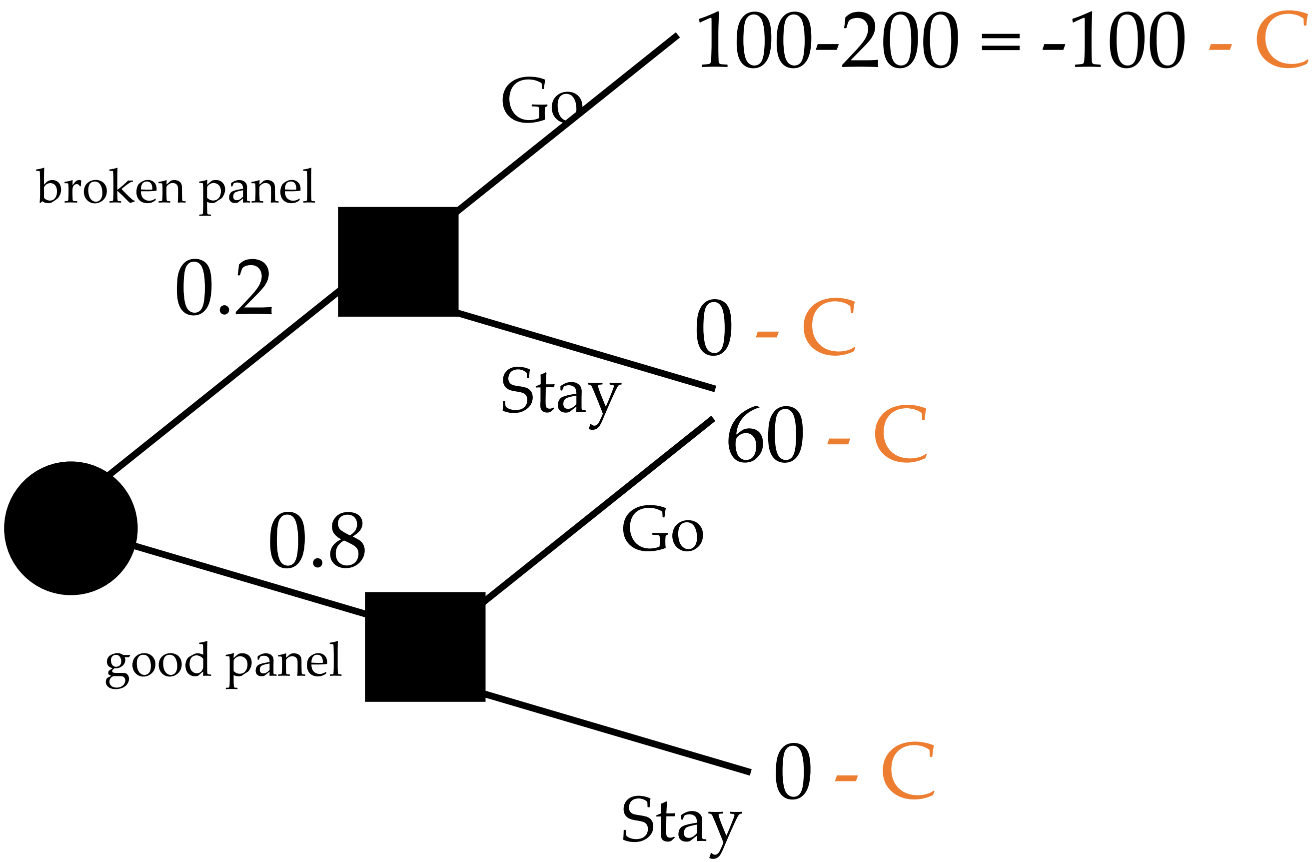
- C ties the expected value with no information



Expected Value of Perfect Information

How much would the agent pay for information of what type of solar panel it is?

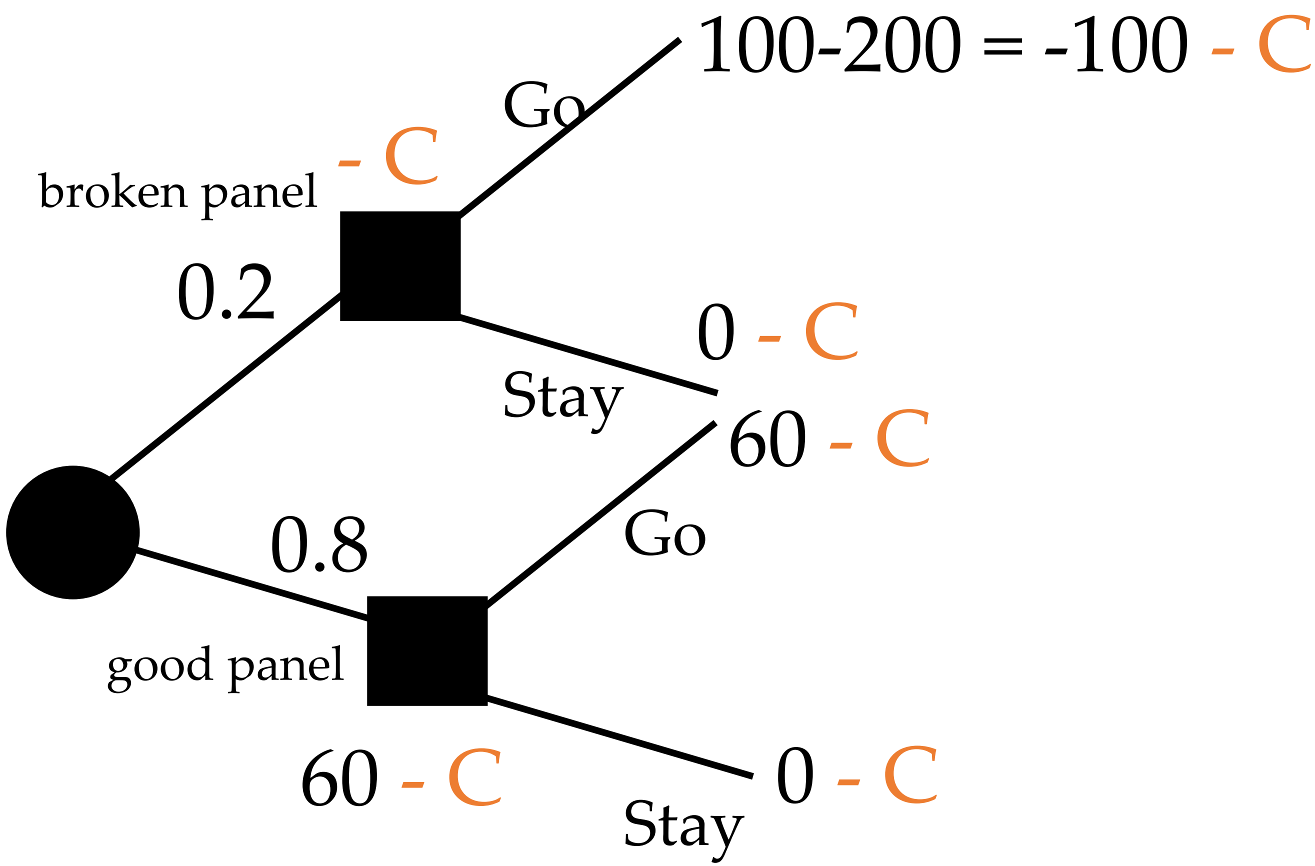
- C ties the expected value with no information



Expected Value of Perfect Information

How much would the agent pay for information of what type of solar panel it is?

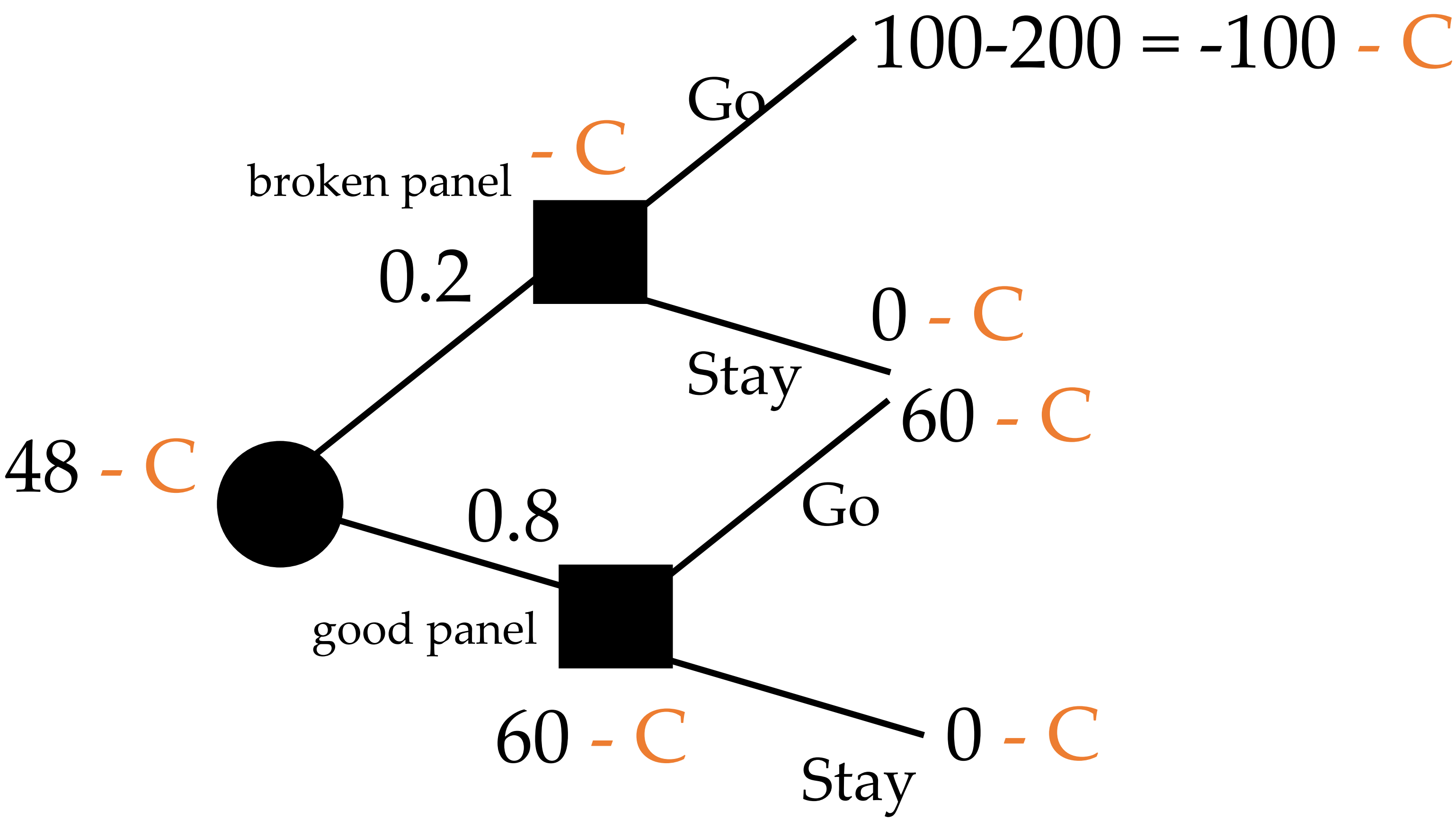
- C ties the expected value with no information



Expected Value of Perfect Information

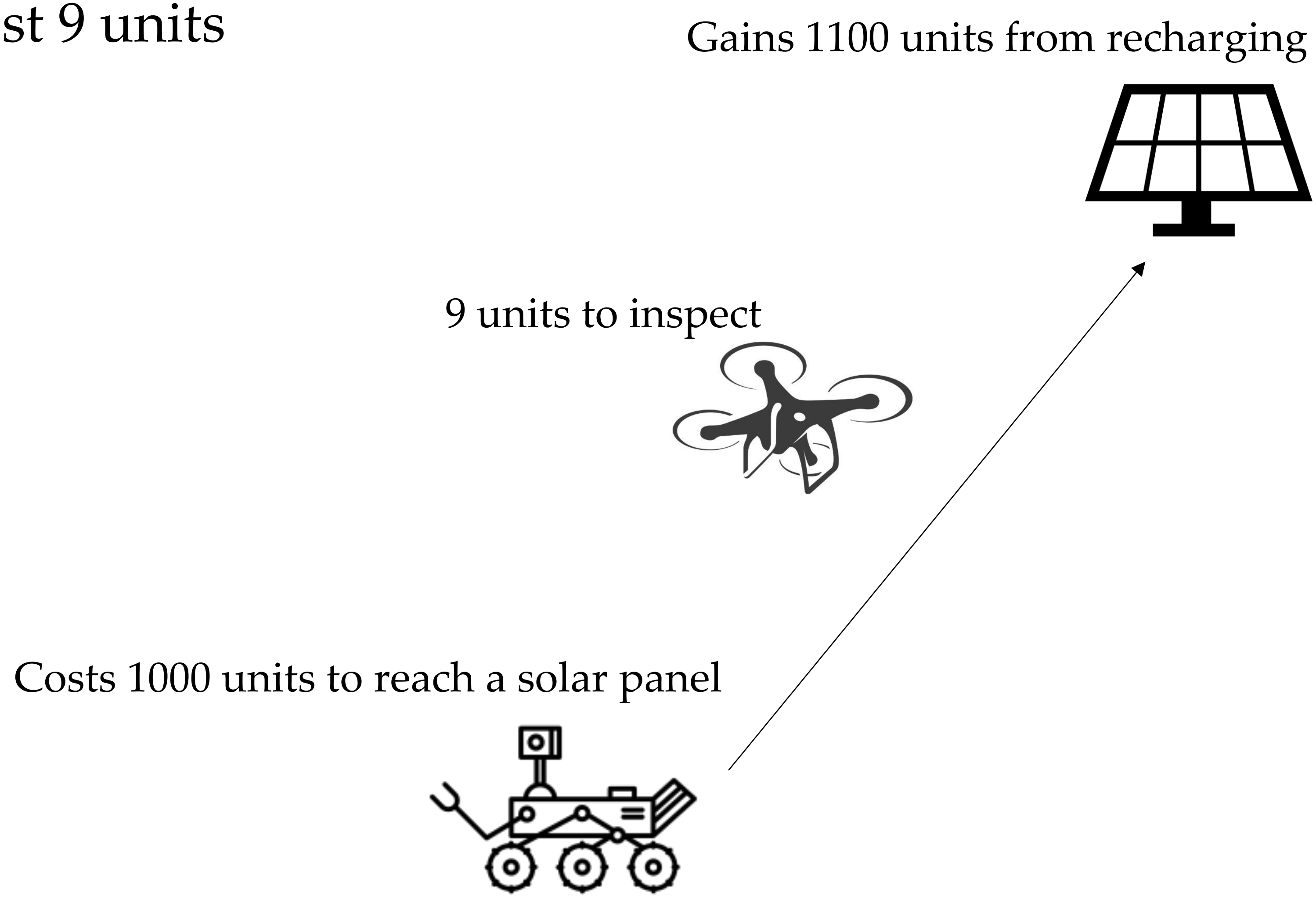
How much would the agent pay for information of what type of solar panel it is?

- C ties the expected value with no information



Inspection?

What if the rover can deploy a **noisy** drone to inspect that will cost 9 units



Inspection?

What if the rover can deploy a drone to inspect that will cost 9 units

$$P(z_t = pass | x_t = good) = 0.9$$

$$P(z_t = fail | x_t = good) = 0.1$$

$$P(z_t = pass | x_t = broken) = 0.4$$

$$P(z_t = fail | x_t = broken) = 0.6$$

Inspection?

What if the rover can deploy a drone to inspect that will cost 9 units

$$P(z_t = pass | x_t = good) = 0.9$$

$$P(z_t = fail | x_t = good) = 0.1$$

$$P(z_t = pass | x_t = broken) = 0.4$$

$$P(z_t = fail | x_t = broken) = 0.6$$

$P(pass) =$

Inspection?

What if the rover can deploy a drone to inspect that will cost 9 units

$$P(z_t = pass | x_t = good) = 0.9$$

$$P(z_t = fail | x_t = good) = 0.1$$

$$P(z_t = pass | x_t = broken) = 0.4$$

$$P(z_t = fail | x_t = broken) = 0.6$$

$$P(pass) = P(pass|good)P(good) + P(pass|broken)(broken)$$

Inspection?

What if the rover can deploy a drone to inspect that will cost 9 units

$$P(z_t = pass | x_t = good) = 0.9$$
$$P(z_t = fail | x_t = good) = 0.1$$
$$P(z_t = pass | x_t = broken) = 0.4$$
$$P(z_t = fail | x_t = broken) = 0.6$$

$$P(pass) = P(pass|good)P(good) + P(pass|broken)(broken)$$
$$= 0.9 * 0.8 + 0.4 * 0.2 = 0.8$$

Inspection?

$$P(z_t = pass | x_t = good) = 0.9$$
$$P(z_t = fail | x_t = good) = 0.1$$
$$P(z_t = pass | x_t = broken) = 0.4$$
$$P(z_t = fail | x_t = broken) = 0.6$$

$$P(broken | pass) =$$

Inspection?

$$P(z_t = pass | x_t = good) = 0.9$$

$$P(z_t = fail | x_t = good) = 0.1$$

$$P(z_t = pass | x_t = broken) = 0.4$$

$$P(z_t = fail | x_t = broken) = 0.6$$

$$P(broken | pass) = \frac{P(pass | broken) P(broken)}{P(pass)}$$

Inspection?

$$P(z_t = pass | x_t = good) = 0.9$$
$$P(z_t = fail | x_t = good) = 0.1$$
$$P(z_t = pass | x_t = broken) = 0.4$$
$$P(z_t = fail | x_t = broken) = 0.6$$

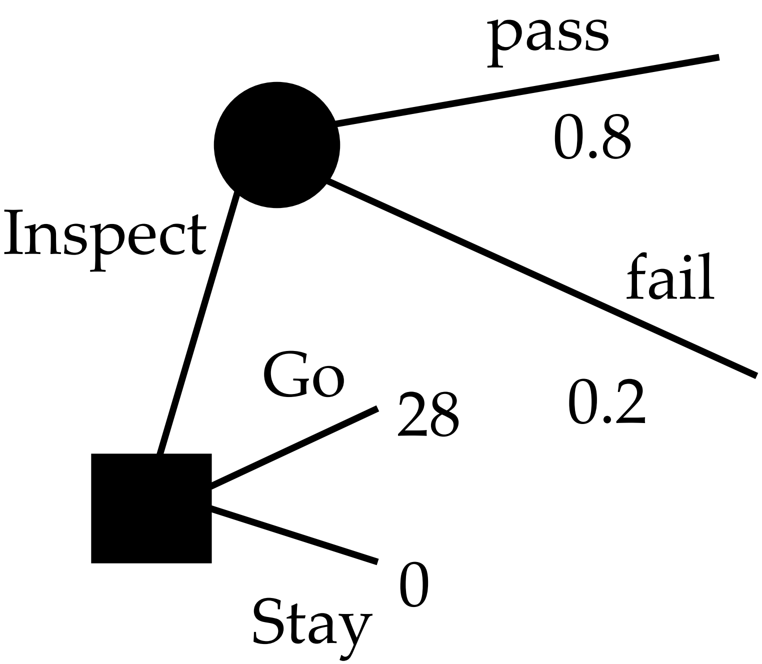
$$P(broken|pass) = \frac{P(pass|broken)P(broken)}{P(pass)}$$
$$= \frac{0.4 * 0.2}{0.8} = 0.1$$

Inspection?

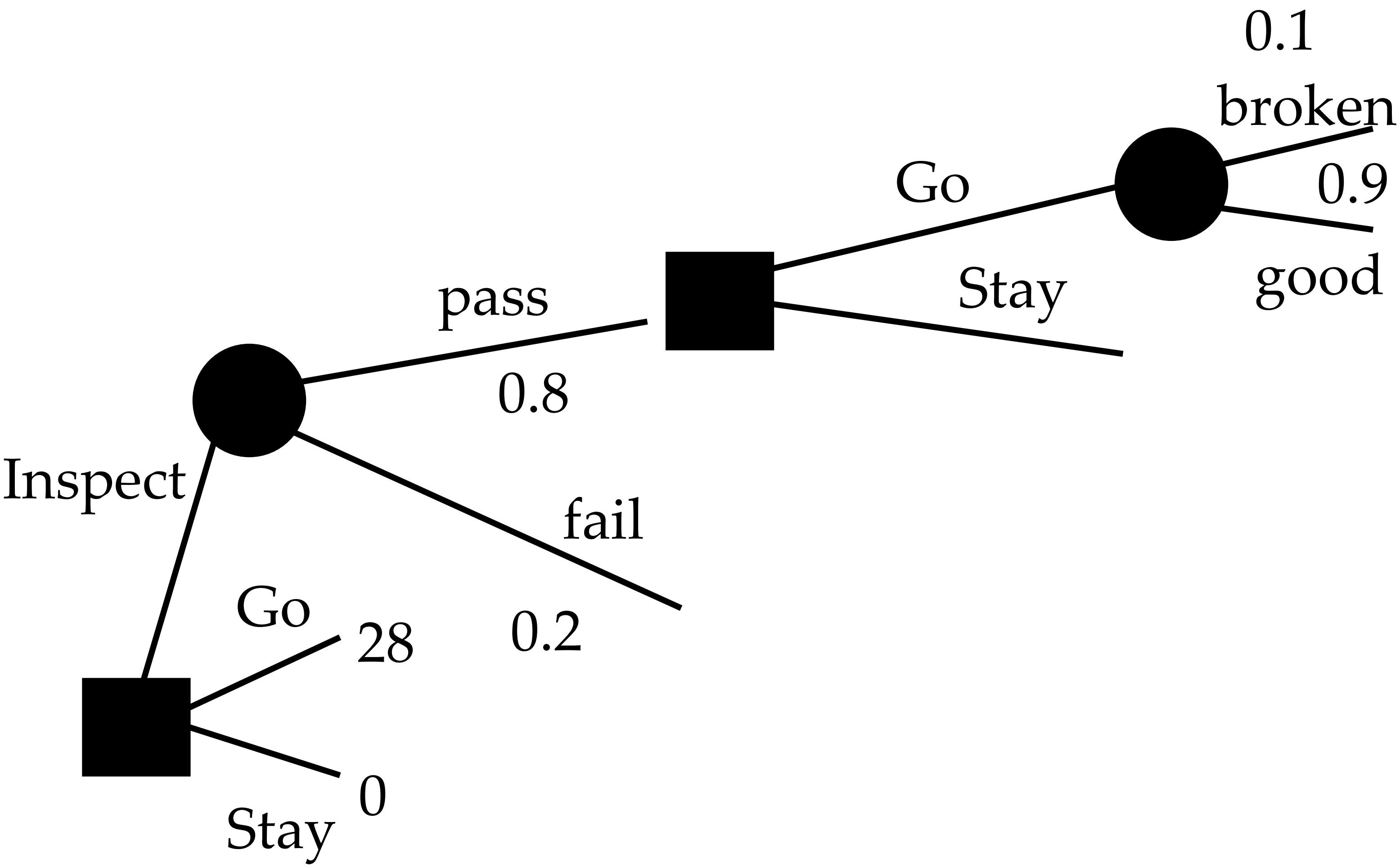
$$P(z_t = pass | x_t = good) = 0.9$$
$$P(z_t = fail | x_t = good) = 0.1$$
$$P(z_t = pass | x_t = broken) = 0.4$$
$$P(z_t = fail | x_t = broken) = 0.6$$

$$P(broken | pass) = \frac{P(pass | broken) P(broken)}{P(pass)}$$
$$= \frac{0.4 * 0.2}{0.8} = 0.1$$
$$P(broken | fail) = \frac{P(fail | broken) P(broken)}{P(fail)}$$
$$= \frac{0.6 * 0.2}{0.2} = 0.6$$

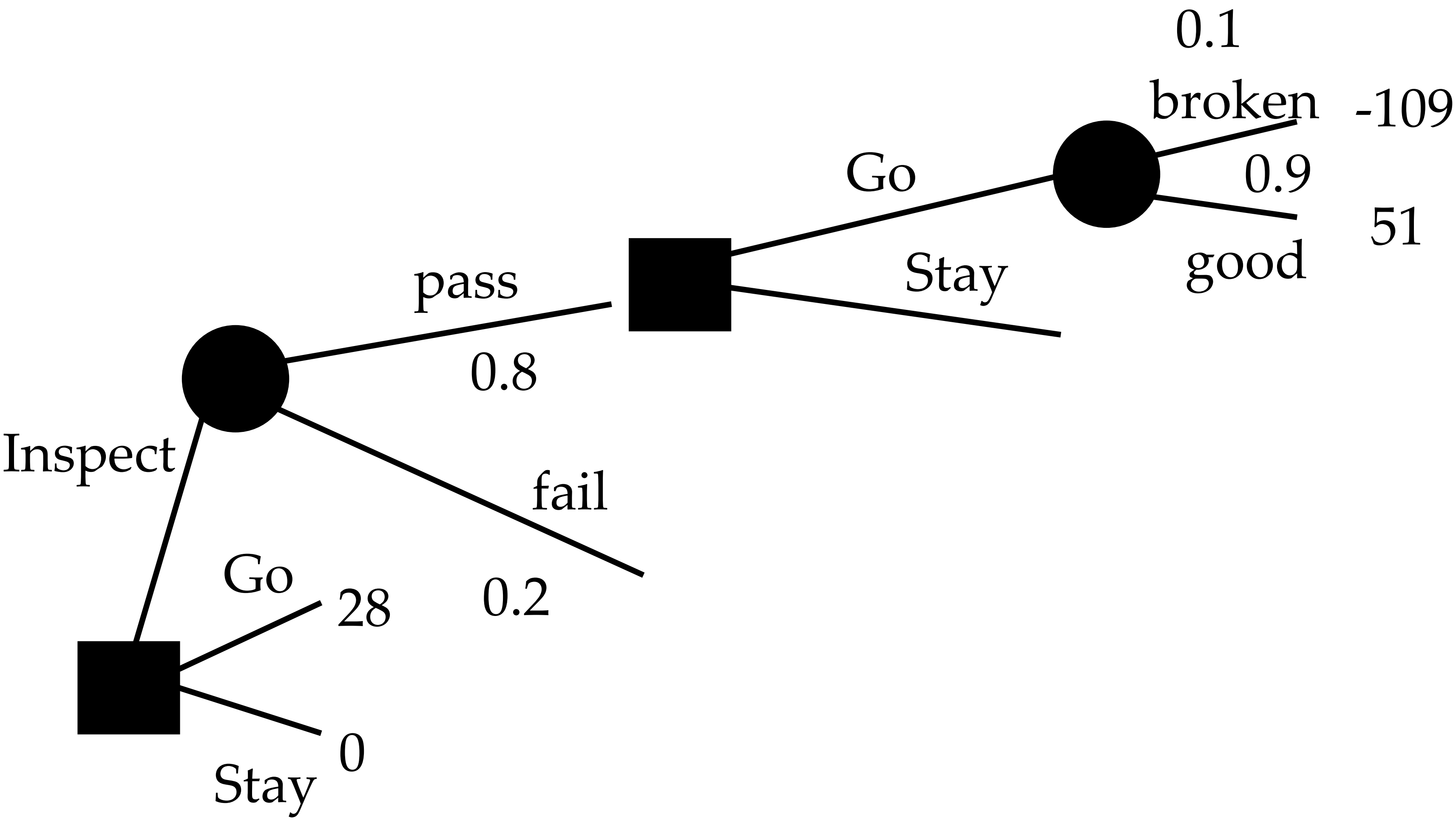
Inspection Tree



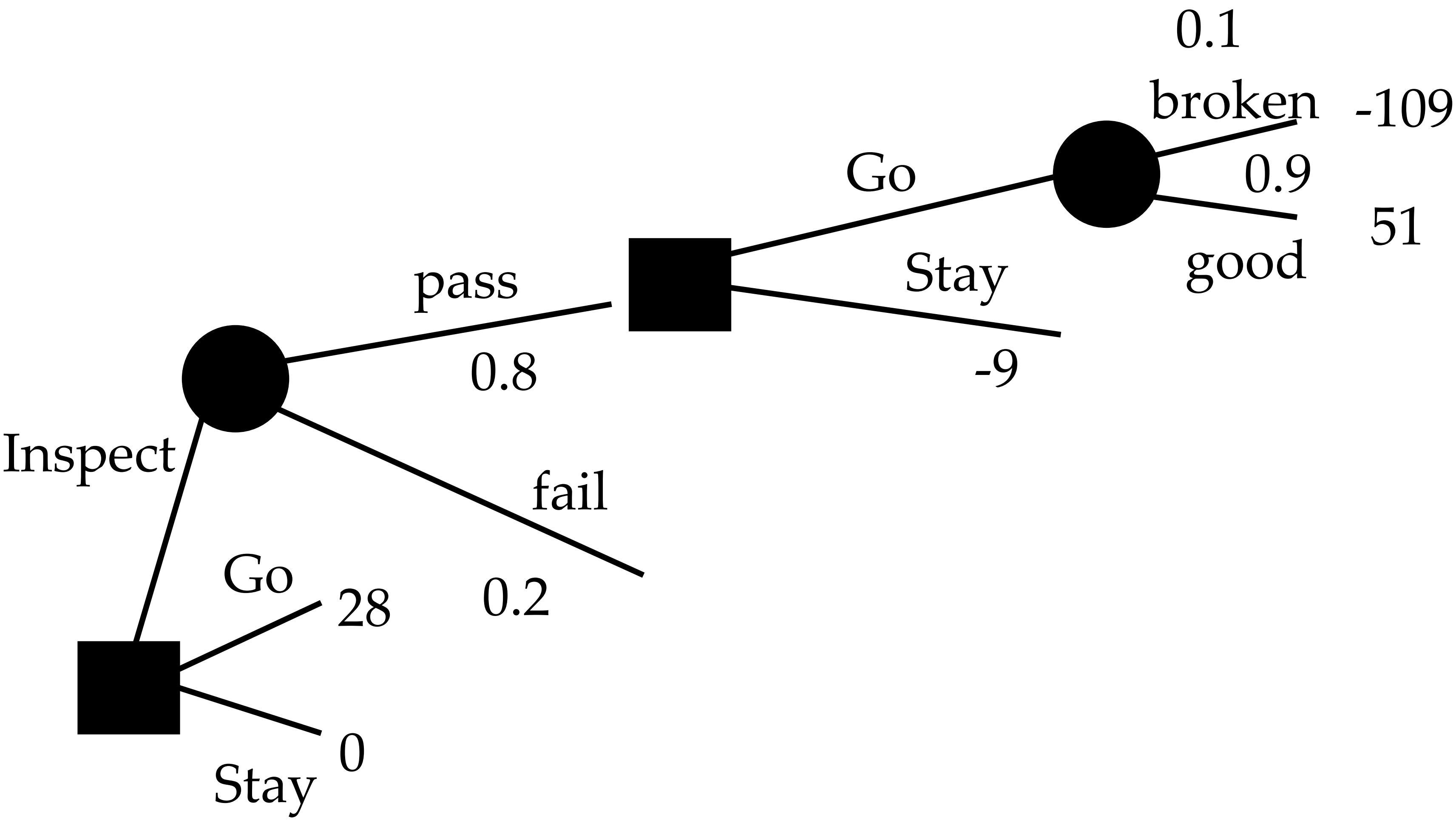
Inspection Tree



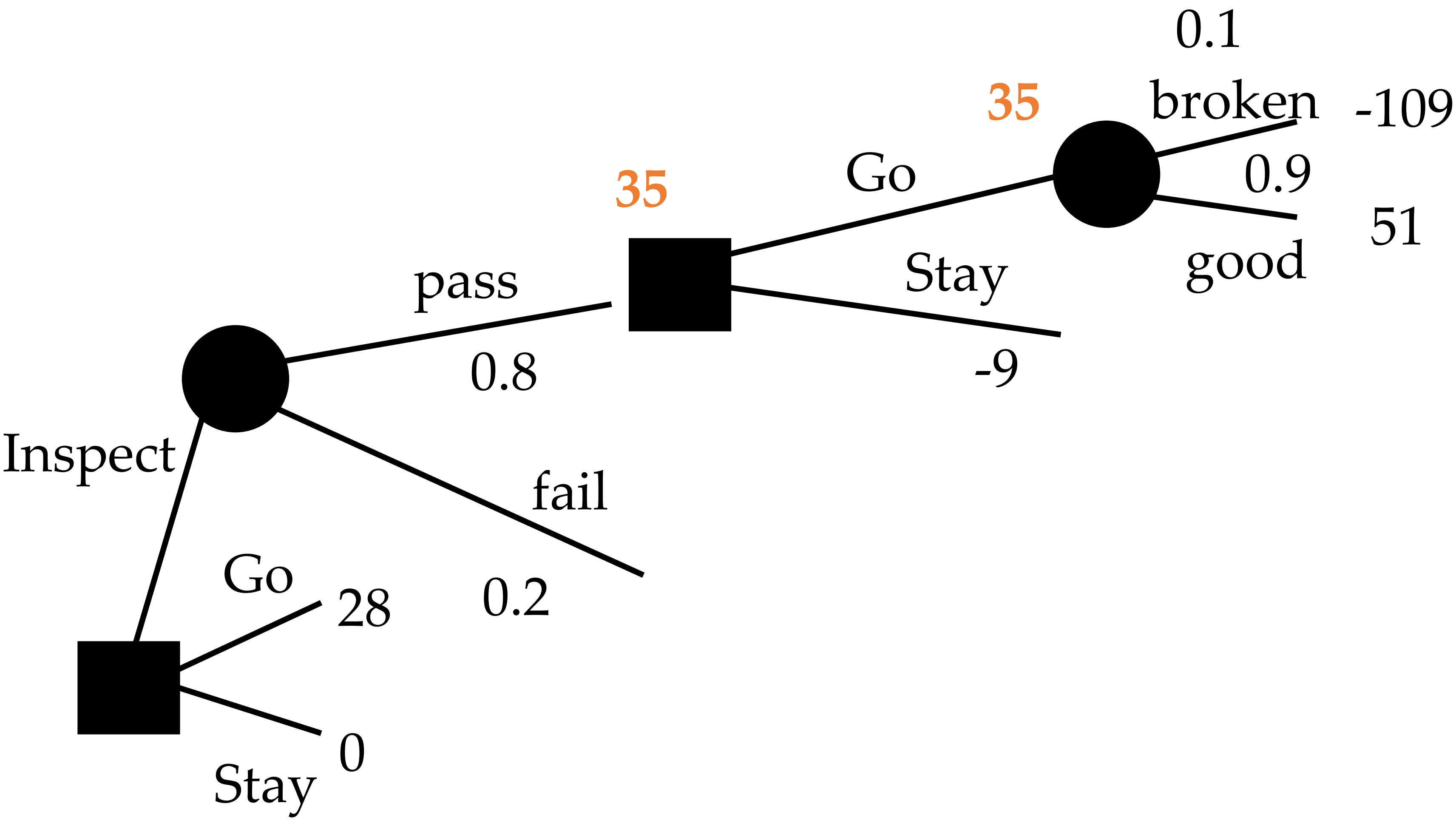
Inspection Tree



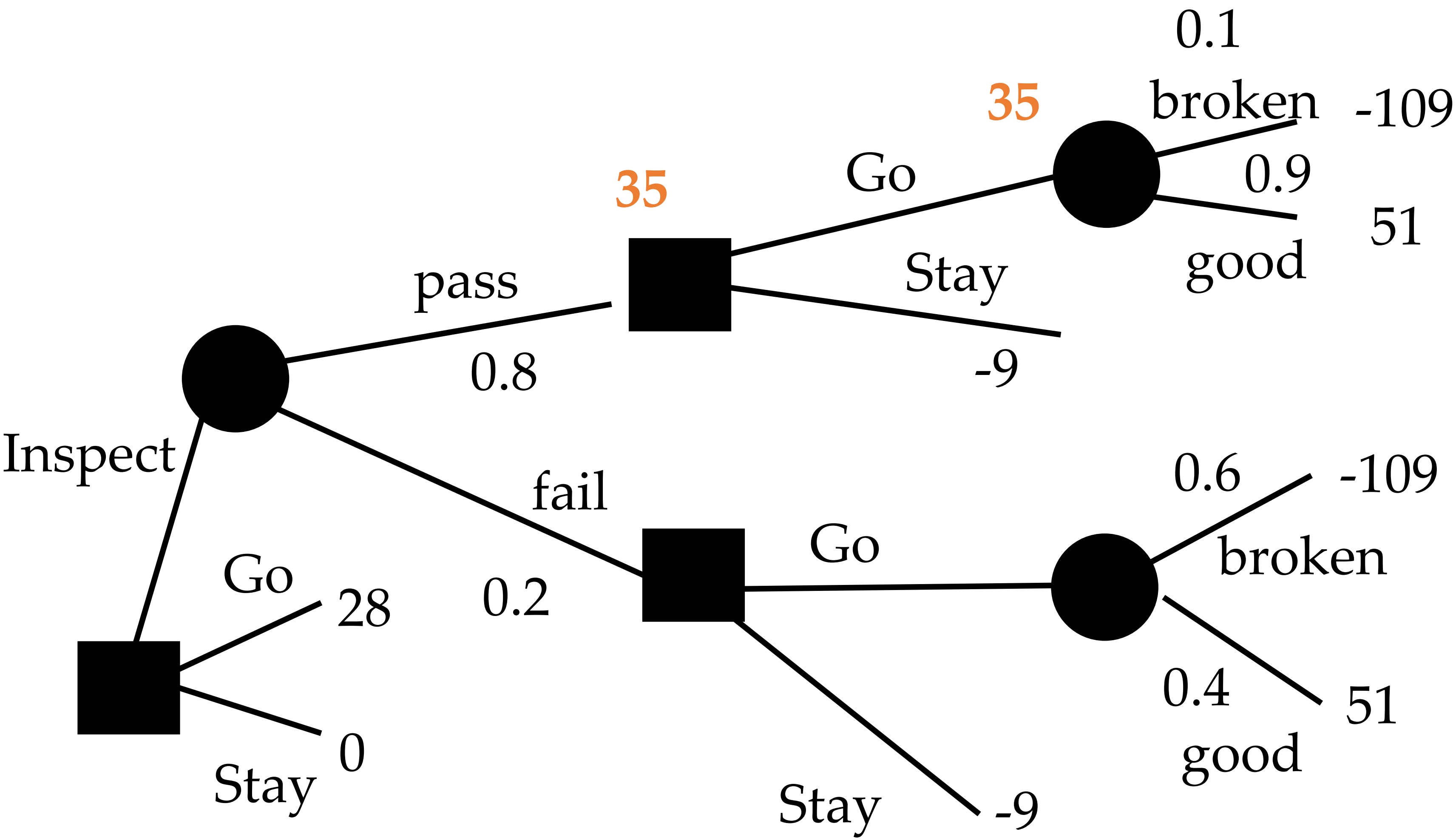
Inspection Tree



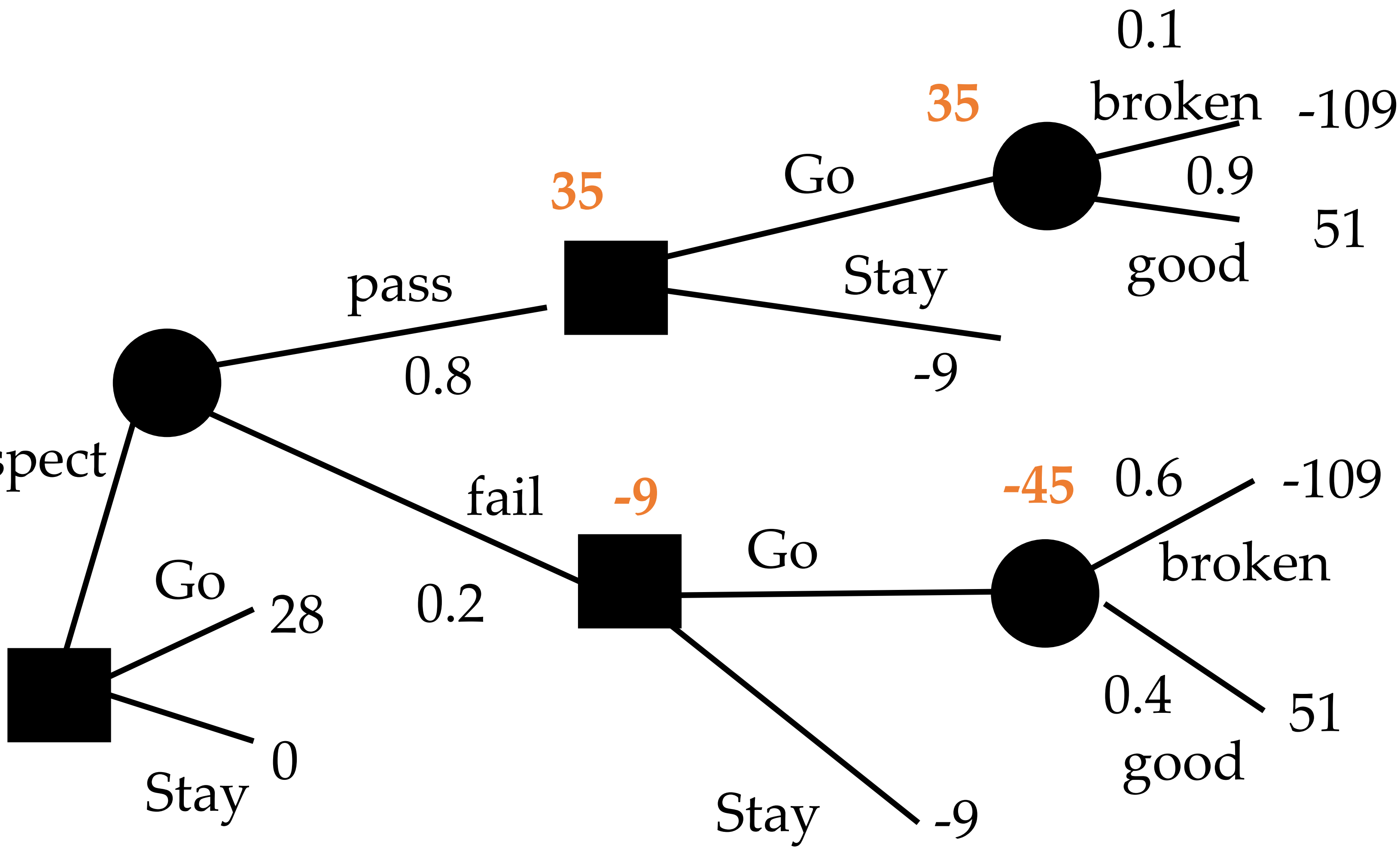
Inspection Tree



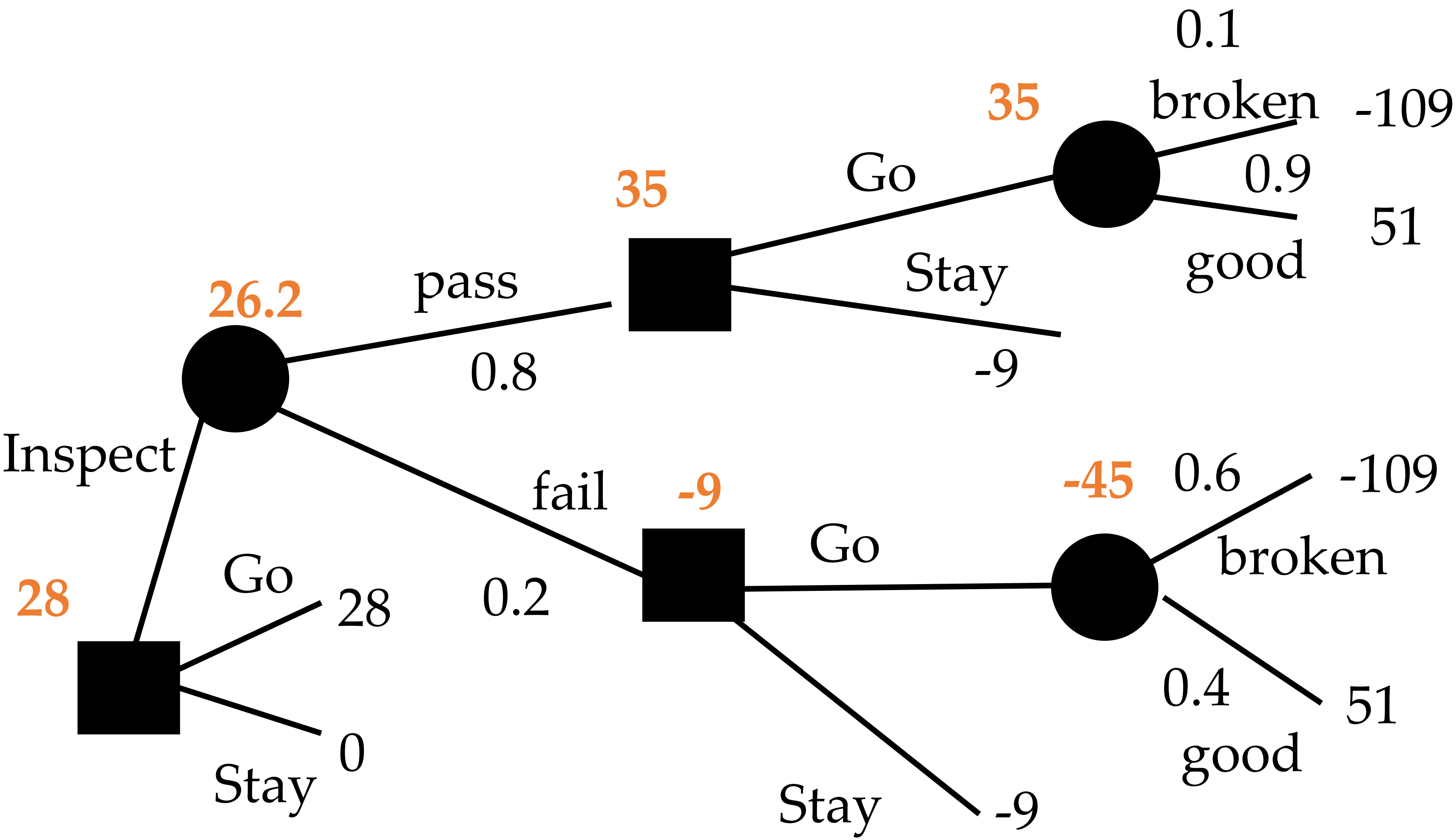
Inspection Tree



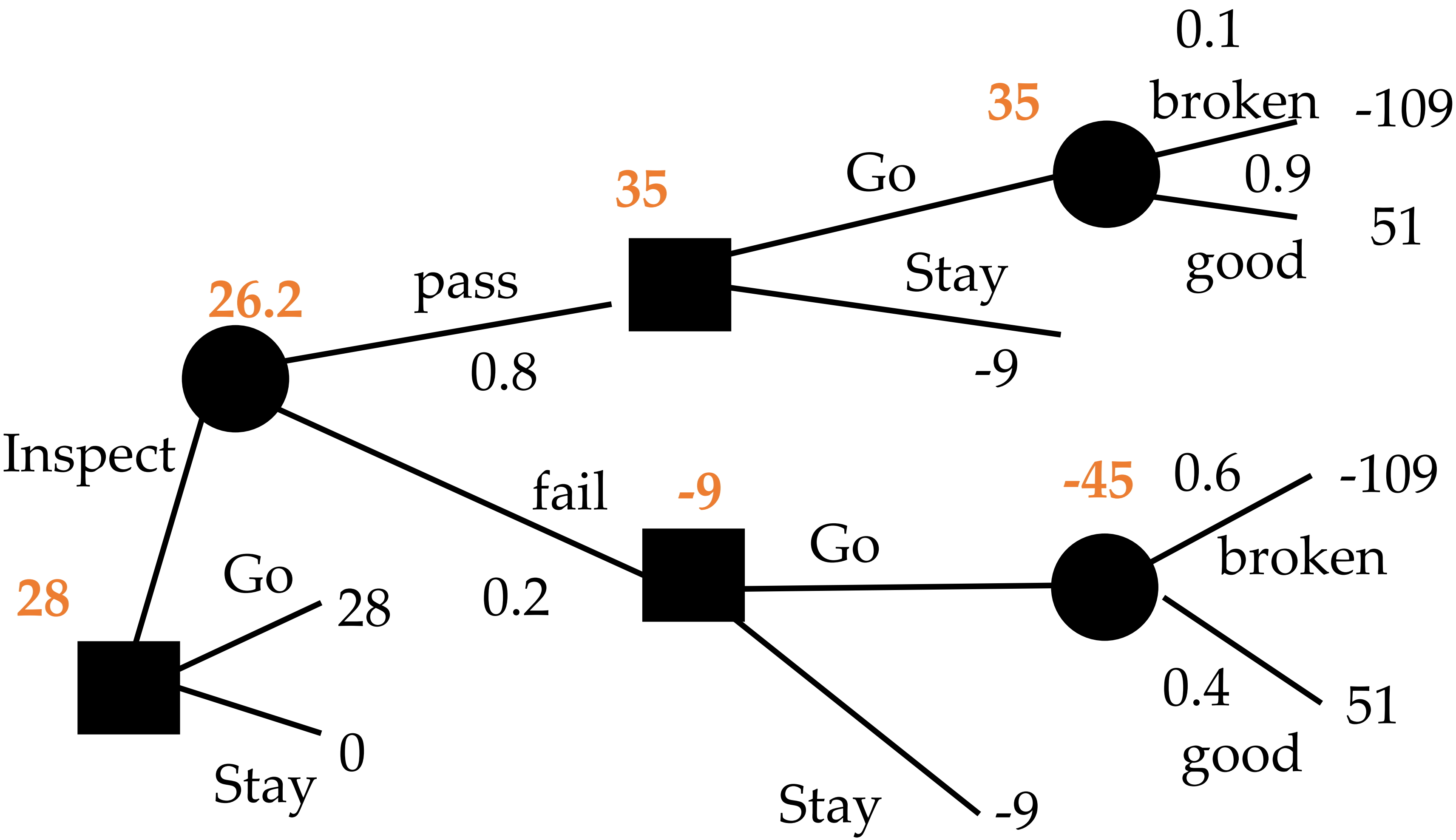
Inspection Tree



Inspection Tree



Inspection Tree



Inspection Tree

Worst case outcome for Inspect: -109, with prob. 0.1
Worst case outcome for Go: -100, with prob. 0.2

A risk-averse agent would probably choose to inspect

USC Viterbi School of Engineering

Risk-Aware Planning

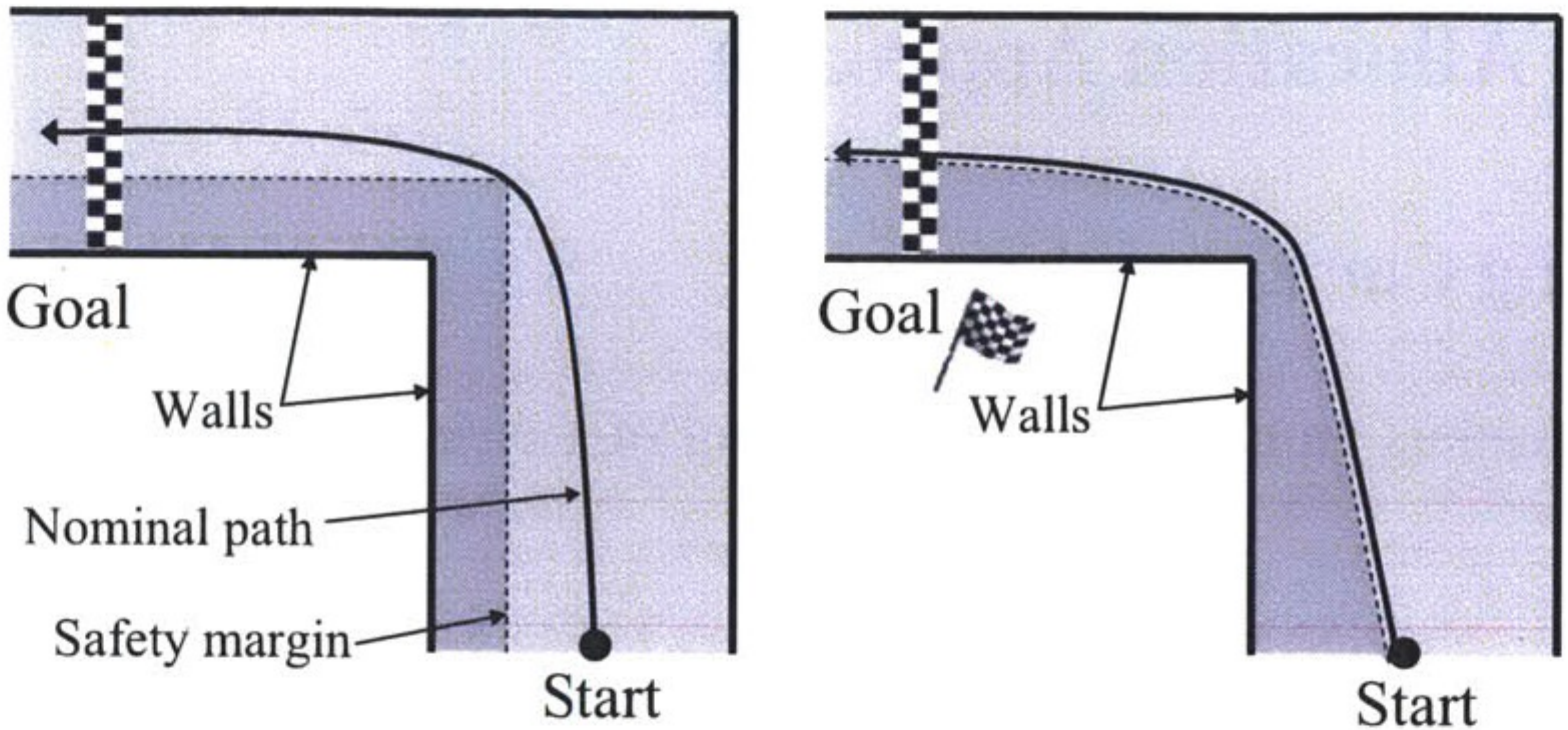
CSCI 545 Introduction to Robotics
Instructor: Stefanos Nikolaidis

[Resources: ”An Efficient Motion Planning Algorithm for Stochastic Dynamic Systems with Constraints on Probability of Failure,” Ono and Williams]

Problem

- Can we find the best path so that the total probability of collision is less than a threshold? (say 0.1%)

Racecar Example



Definitions

- Feasible Region: R_t

Definitions

- Feasible Region: R_t
- Dynamics: $x_{t+1} = g(x_t, u_t, w_t)$

Definitions

- Feasible Region: R_t
- Dynamics: $x_{t+1} = g(x_t, u_t, w_t)$
- Nominal states: $\bar{x}_{t+1} = g(\bar{x}_t, u_t, 0)$

Definitions

- Feasible Region: R_t
- Dynamics: $x_{t+1} = g(x_t, u_t, w_t)$
- Nominal states: $\bar{x}_{t+1} = g(\bar{x}_t, u_t, 0)$
- Probability of Failure:

$$P_{fail} = P[(x_1 \notin R_1) \cup (x_2 \notin R_2) \cup ... \cup (x_t \notin R_T)]$$

Definitions

- Feasible Region: R_t
- Dynamics: $x_{t+1} = g(x_t, u_t, w_t)$
- Nominal states: $\bar{x}_{t+1} = g(\bar{x}_t, u_t, 0)$
- Probability of Failure:

$$P_{fail} = P[(x_1 \notin R_1) \cup (x_2 \notin R_2) \cup \ldots \cup (x_t \notin R_T)]$$

- Chance constraint over mission: $P_{fail} \leq \delta$
- Objective function: $J = h(\bar{x}_{1:T}, \bar{u}_{1:T})$

Constraint Decomposition

$$P_{fail,1} \leq \delta_1$$

$$P_{fail,2} \leq \delta_2$$

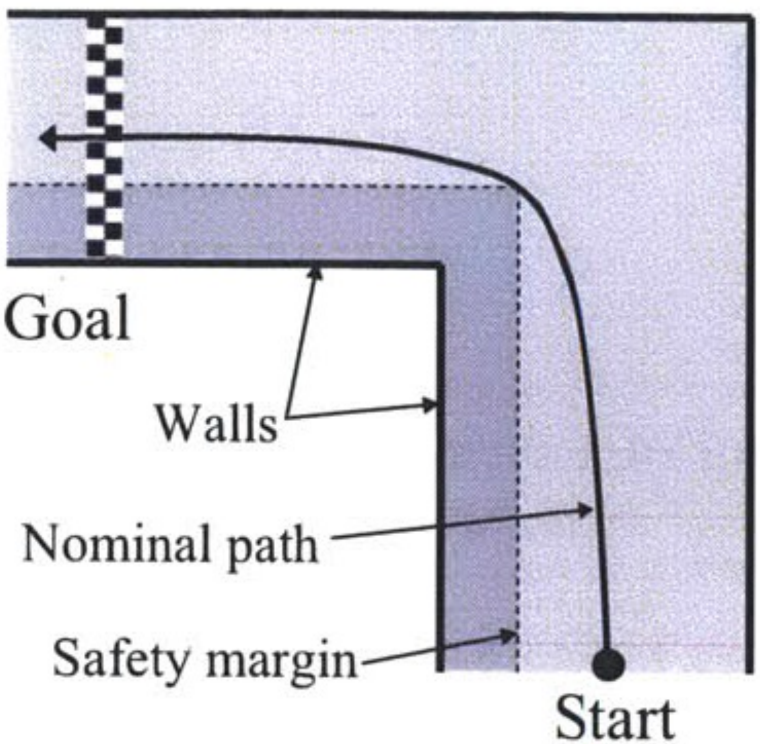
$$\vdots$$

$$P_{fail,T} \leq \delta_T$$

$$\sum_{t=1}^{T} \delta_t \leq \delta$$

Safety Margin

$$P(x_t \notin R_t) \leq \delta_1 \rightarrow \bar{x}_t \in (R_t - M_t)$$



Iterative Risk Allocation Algorithm

Lower Stage Optimization:

Convert chance constraints $\delta_1, ..., \delta_T$ to safety margin constraints

Find path that minimizes $J = h(\bar{x}_{1:T}, \bar{u}_{1:T})$

Iterative Risk Allocation Algorithm

Upper Stage Optimization:

Get the timesteps that optimal path "touches" feasible regions

Update $\delta_1, ..., \delta_T$ so that you increase probability of collision for these timesteps and decrease for the rest

Racecar Example