CS 7180: Behavioral Modeling and Decision-making in AI

Abductive Reasoning

Prof. Amy Sliva

November 16, 2012

How do we reason about behavior

- From what we know about the world, what can happen next?
  - Know *best_move*(*a*) → *play*(*a*), *best_move*(*QUEEN_TO_E3*), so conclude *play*(*QUEEN_TO_E3*)
  - **Deduction—determine the conclusion**

- From observations, construct a general behavioral rule
  - Observe *best_move*(*QUEEN_TO_E3*) and *play*(*QUEEN_TO_E3*), so add *best_move*(*a*) → *play*(*a*) as a plausible rule
  - **Induction—determine the rule**

- Explain possible causes of behavior
  - Know *best_move*(*a*) → *play*(*a*) and observe *play*(*QUEEN_TO_E3*) so infer *best_move*(*QUEEN_TO_E3*) as a possible explanation
  - **Abduction—determine the preconditions**

Abductive reasoning

- Process of forming an explanatory hypothesis from a set of observations
  - **Inference to the best explanation**

- Hypothetical reasoning has many applications
  - Diagnostics
  - Image processing
  - Planning
  - Natural language processing
  - Temporal and probabilistic reasoning
  - …

- We use it to develop **policies** or **counterstrategies**
  - Hypothetical explanations of desirable behavior
  - How we can get another agent to act in a more favorable way

What is the best explanation?

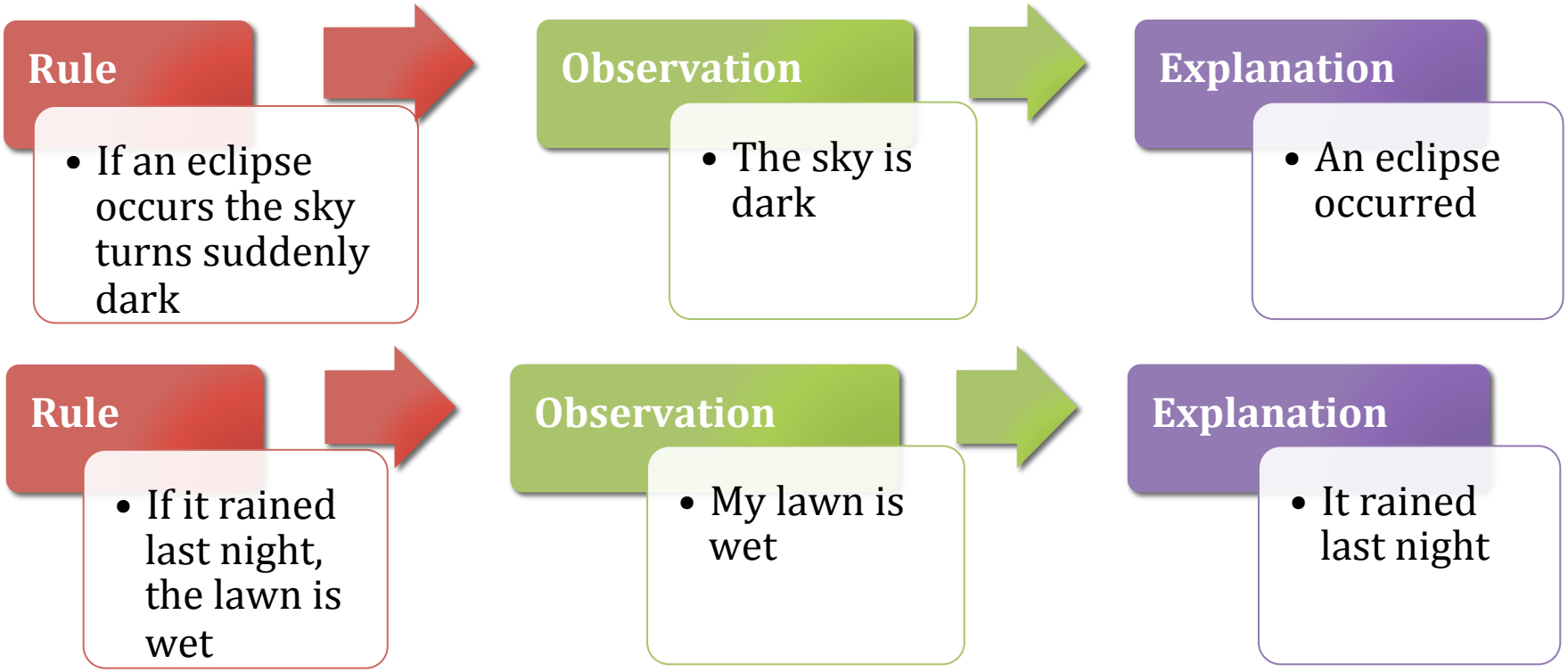
- Charles Sanders Pierce (1839-1914)
  - **"Abduction is no more nor less than guessing"**

Rule
- All beans in the bag are white

Observation
- The beans are white

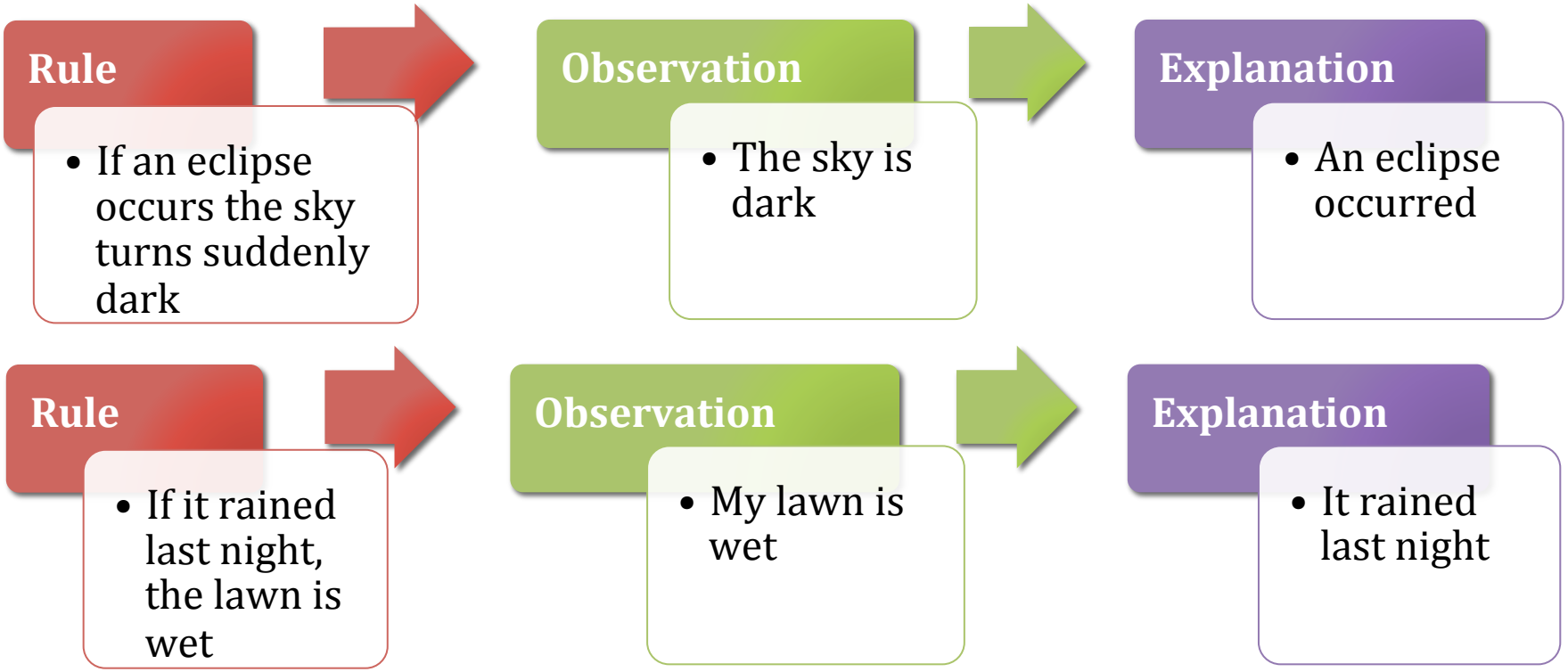
Explanation
- Therefore, these beans are from the bag

Is abductive inference correct?



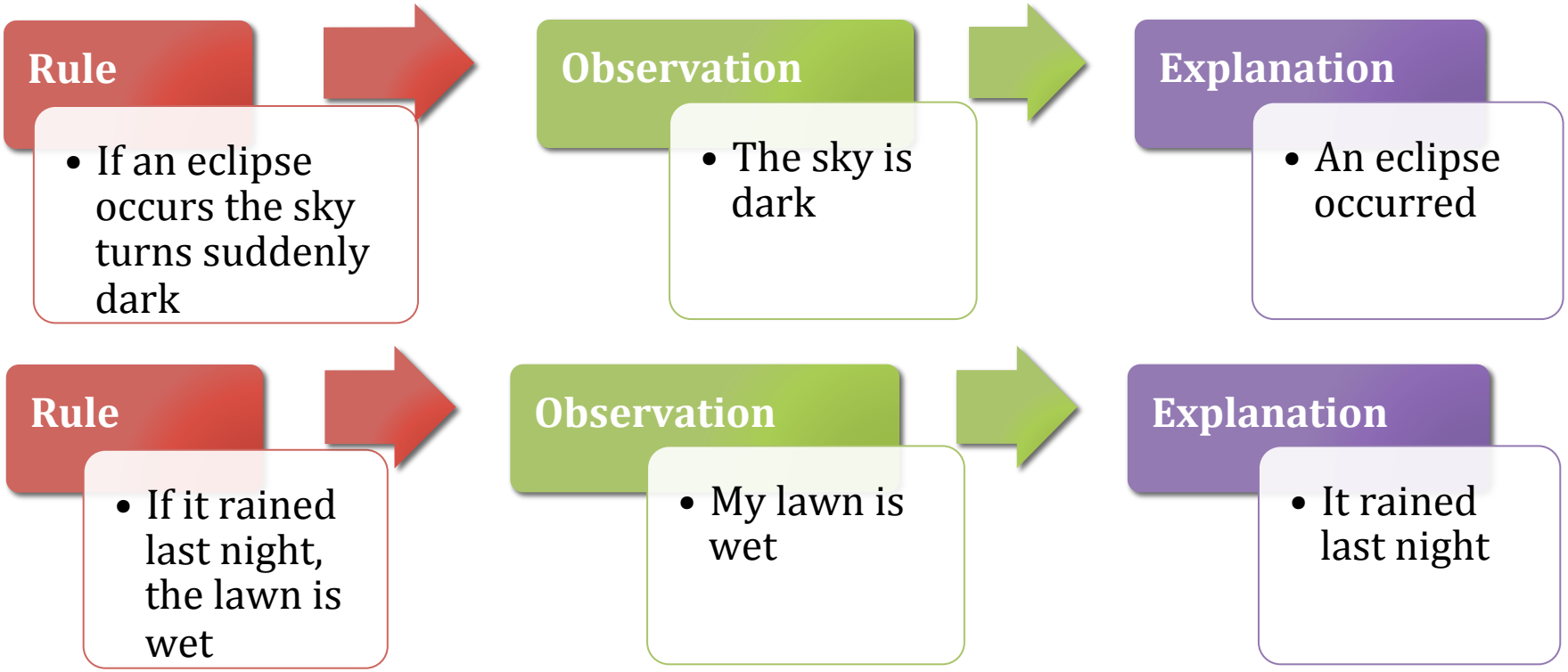
- Well…?

Is abductive inference correct?



- "Induction and abduction are fallible forms of reasoning. Their conclusions are susceptible to retraction."

Is abductive inference correct?



- "Success of our guesses far exceed that of random luck…"

Why is abductive reasoning useful?

- Theory formation
  - When Newton saw the apple falling down, he must have done an abductive inference and came up with the theory of gravity!
  - A possible thought process:

Observation

- Apple fell

Rule

- If Earth pulled everything toward is, then of course the apple too would fall down

Explanation

- So, Earth is pulling everything toward it

Stages in mental process of abduction

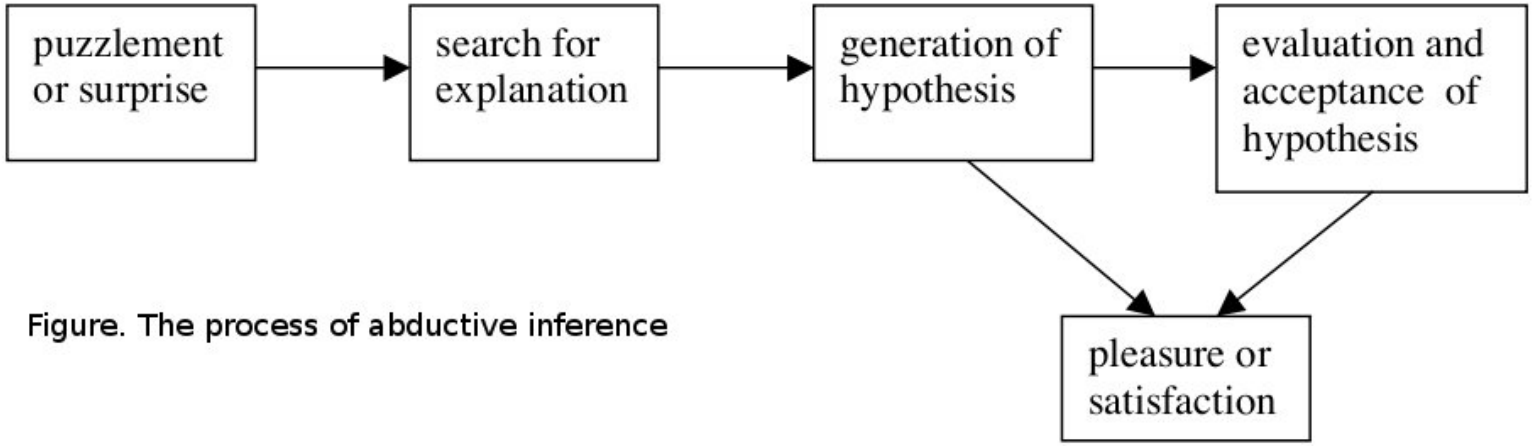


Figure. The process of abductive inference

Source- Abductive Inference: From Philosophical Analysis to Neural Mechanisms by Paul Thagard(University of Waterloo)

- Abduction is essentially the starting point of all research!

Types of abductive reasoning

- Logic approach
  - Defining a logical theory based on a set of sentences (explanations) that describe the observations
  - Finding best explanation (policy) to for a particular goal
- Set-Cover approach
- Probabilistic approach
  - Finding most probable explanation for observations
  - Can be combined with logic

Types of abductive reasoning

- **Logic approach**
  - Defining a logical theory based on a set of sentences (explanations) that describe the observations
  - Finding best explanation (policy) to for a particular goal
- Set-Cover approach
- **Probabilistic approach**
  - Finding most probable explanation for observations
  - Can be combined with logic

Logic-based approach to abduction

- A logical theory (KB) T defined over logical language L
- Subset of sentences $\varepsilon \in L$ called **abducibles**
  - **Possible hypotheses**

- Abductive inference problem (T, ε, O)
  - For an **observation** O find some set of sentences H s.t.
  - $T \cup H$ is consistent
  - $T \cup H \models O$
  - $H \in \varepsilon$

What is the "best" explanation?

- Usually involves some concept of **minimality**
  - **Simpler** explanations are better
  - H is a minimal abductive explanation for O iff there is no other explanation H' s.t. H' is simpler than H

- Examples of simplicity
  - $|H| \leq |H'|$ for all other solutions H'
  - No other solution H' s.t. $H' \subseteq H$
  - No other solution H' s.t. forall $f \in H'$, there is some $g \in H$ s.t. $g \models f$

- **Domain dependent**
  - May be many parsimonious solutions, how do we choose one?
  - Need additional domain knowledge or specific framework for ranking (can also use probabilities)

Causal logic theory

- Limited expressiveness, but good for diagnostic domains

- Causal abductive **framework** is triple (C, E, T)
  - $C \in L$ is a set of atomic sentences that are **causes**
  - $E \in L$ is a set of atomic sentences that are **effects**
  - T is the causal theory—the logic KB

- **Abductive explanation** for framework (C, E, T) of observations $O \subseteq E$ is a hypothesis H s.t.
  - $H \subseteq C$
  - $T \cup H$ is consistent
  - $T \cup H \models O$
  - H is subset minimal

Causal abduction example

Suppose the following KB regarding animal behavior:

(1) $\forall x(bird(x) \wedge \neg ab(x) \rightarrow flies(x))$

(2) $\forall x(ufo(x) \rightarrow flies(x))$

(3) $\forall x(penguin(x) \vee ostrich(x) \rightarrow ab(x))$

(4) $\forall x(songbird(x) \rightarrow bird(x))$

(5) $\forall x(songbird(x) \rightarrow eats\ insects(x))$

(6) $\forall x(frog(x) \rightarrow eats_insects(x))$

(7) $\forall x(frog(x) \rightarrow green(x) \wedge croaks(x))$

(8) $\forall x(frog(x) \rightarrow ab(x))$

Causal abduction example

Have a causal model (C, E, T) where

T = KB on previous slide

C = {*frog(x), songbird(x), bird(x), ostrich(x), penguin(x)*}

E = {*flies(x), green(x), croaks(x), eats insects(x)*}

Have set of **observations** O = {¬*flies(F), croaks(F)*}

Causal abduction example

Have a causal model (C, E, T) where

T = KB on previous slide

C = {*frog(x), songbird(x), bird(x), ostrich(x), penguin(x)*}

E = {*flies(x), green(x), croaks(x), eats insects(x)*}

Have set of observations O = {¬*flies(F), croaks(F)*}, then

Possible **explanation** H= {*frog(F)*}

Causal abduction example

Have a causal model (C, E, T) where

T = KB on previous slide

C = {*frog(x), songbird(x), bird(x), ostrich(x), penguin(x)*}

E = {*flies(x), green(x), croaks(x), eats insects(x)*}

Have set of observations O = {$\neg$*flies(F), croaks(F)*}, then

Possible explanation H= {*frog(F)*}

- *frog(F)* is **consistent** with T
- $T \cup$ *frog(F)* $\models$ $\neg$*flies(F)* $\wedge$ *croaks(F)*, and
- *frog(F)* is **subset-minimal**

Abductive logic programming

- Extension to logic programming with abduction

- Logic programming abductive framework (P, ε, IC)
- Separates theory in two parts
  - Normal logic program P to identify $H \subseteq \varepsilon$ (abducibles)
    - Rules of the form $A \leftarrow L_1 \wedge \ldots \wedge L_n$ where $L_1, \ldots, L_n$ are literals and A is an atom
  - **Integrity constraints** IC to filter set of possible candidates

- **Abductive explanation** for framework (P, ε, IC) of observations O is a hypothesis H s.t.
  - $H \subseteq \varepsilon$
  - $P \cup H$ is consistent
  - $P \cup H \models O$
  - H does not violate IC

Abductive logic programming example

- Logic Program P

  $WetGrass \leftarrow Rain.$

  $WetGrass \leftarrow Sprinkler.$

  $Sunny.$

- Integrity constraints IC

  $false \leftarrow Rain \wedge Sunny.$

- Observation O

  $WetGrass$

- **What is the best explanation?**

Abductive logic programming example

- Logic Program P

  $WetGrass \leftarrow Rain.$

  $WetGrass \leftarrow Sprinkler.$

  $Sunny.$

- Integrity constraints IC

  $false \leftarrow Rain \wedge Sunny.$

- Observation O

  $WetGrass$

- ***Sprinkler***

Abductive logic programming example

- Logic Program P

  $WetGrass \leftarrow Rain.$

  $WetGrass \leftarrow Sprinkler.$

  $Sunny.$

- Integrity constraints IC

  $false \leftarrow Rain \wedge Sunny.$

- Observation O

  $WetGrass$

- ***Sprinkler***

IC means *Rain* cannot be an explanation since it is *Sunny.*

Abductive logic programming example

- Logic Program P

  $WetGrass \leftarrow Rain.$

  $WetGrass \leftarrow Sprinkler.$

  $Sunny.$

- Integrity constraints IC

  $false \leftarrow Rain \wedge Sunny.$

- Observation O

  $WetGrass$

- ***Sprinkler***

IC means *Rain* cannot be an explanation since it is *Sunny.*

Only possible explanation is *Sprinkler*

Abduction in probabilistic logic

- Use abduction to determine explanation w.r.t. integrity constraints and a **probability interval**

- Find way to **reach** a (feasible) state s.t. logic theory is consistent and entails that goal will be true within some probability bounds
  - Use to determine **policies** or **strategies**—what can we do to induce certain behavior?

- Use action probabilistic logic programs for this abduction problem

Representing behavior in PLPs

- **Action probabilistic** (*ap*) programs
  - Specialized variants of PLPs for behavior
  - Bipartite structure—atoms divided into **action** and **state**

- Only annotate action atoms, and only in the head
  - Like a conditional probability

- *ap*-rule—$F{:}[l,u] \leftarrow B_1 \wedge \ldots \wedge B_n$

$$riot(activists){:}\ [0.5,0.7] \leftarrow elections(Egypt)$$

"Given there are elections in Egypt, activists will riot with a probability of 0.5 to 0.7"

- Any PLP (i.e., even one with actions in the body) can be written in this form using **fixpoint** procedure
  - Iterate to find tightest bounds for each atom

Representing behavior in PLPs

- **Action probabilistic** (*ap*) programs
  - Specialized variants of PLPs for behavior
  - Bipartite structure—atoms divided into **action** and **state**

- Only annotate action atoms, and only in the head
  - Like a conditional probability

- *ap*-rule—$F{:}[l,u] \leftarrow B_1 \wedge \ldots \wedge B_n$

$$riot(activists){:}\ [0.5,0.7] \leftarrow elections(Egypt)$$

"Given there are elections in Egypt, activists will riot with a probability of 0.5 to 0.7"

- Any PLP (i.e., even one with actions in the body) can be written in this form using **fixpoint** procedure
  - Iterate to find tightest bounds for each atom

What behavioral rules apply in a state?

- Define possible **worlds** only over **action** atoms
- Define **states** over state atoms
- *ap*-program models behavior taking **context** into account
  - Conditional probabilities (with ranges) of actions given a context

kidnap: [0.35, 0.45] ← interOrganizationConflicts.
kidnap: [0.60, 0.68] ← unDemocratic ∧ internalConflicts.
armed attacks: [0.42, 0.53] ← typeLeadership(strongSingle) ∧
orgPopularity(moderate).
armed attacks: [0.93, 1.0] ← statusMilitaryWing(standing).

What behavioral rules apply in a state?

- For an *ap*-program, the **reduction** w.r.t. a state *s* is a reduced PLP containing only the rules that apply in *s*
  - Action is **applicable** if body of the rule is true in the state

kidnap: [0.35, 0.45] ← interOrganizationConflicts.
kidnap: [0.60, 0.68] ← unDemocratic ∧ internalConflicts.
armed attacks: [0.42, 0.53] ← typeLeadership(strongSingle) ∧ orgPopularity(moderate).
armed attacks: [0.93, 1.0] ← statusMilitaryWing(standing).

Linear constraints from a PLP

- Derive a set of **linear constraints** where p_i is the probability of world w_i
  - $l \leq \sum_{wi \mapsto F} p_i \leq u$ for each rule $F{:}[l,u]$ s.t. $B_1 \wedge \ldots \wedge B_n$ is true
  - $\sum_{wi} p_i = 1$

- Use constraints to solve entailment queries
- Can also solve for the most probable world
  - Minimize (or maximize) p_i w.r.t. constraints

Probabilistic logic abduction with reachability

- Probabilistic logic abductive framework (Π, *reach*, *s*, G:[α,β])
  - Probabilistic logic program Π
  - Reachability predicate *reach*
    - **Integrity constraints** telling us which states we can get to from *s*
    - True if there is some path of states from *s* to *s'*
  - Current (or hypothetical) state of the world *s*
  - Goal formula G:[α,β]

- **Abductive solution** for (Π, *reach*, *s*, G:[α,β]) is a state *s'* s.t.
  - *reach*(*s*,*s'*)
  - $\Pi_{s'} \models G$:[α,β]

- Finds **sequence** of states from *s* to *s'* *s.t.* reduction of behavioral model in *s'* entails the goal in the given probability range

How do we find reachable states?

- **Reachability** is a general concept
  - From the current state, can we end up at the goal?
  - Considers actions (abducibles) that can be taken—define preconditions and effects
    - **What does this type of solution remind you of?**

- Can define **syntactically** in terms of logic rules
  - Add constraints on reachability
  - $F \rightarrow G$ iff $unReach(s_1,s_2)$ $s_1 \vDash F \wedge s_2 \vDash G$

- Use constraints when solving for reachable states

Probabilistic logic with abducibles

- Formulate similar to traditional logic abduction

- Probabilistic logic abductive framework (Π, ε ,G:[α,β])
  - Action probabilistic logic program Π
  - Abducibles ε
    - Set of possible actions we can take
  - Goal formula G:[α,β]

- Abducibles are subset of **state** variables in bipartite *ap*-program structure
  - How can we change the state of the world to **influence** actions?
- Goal is behavior we want to achieve (within some probability)
- Can add **integrity constraints** to represent **cost**, **feasibility**, etc.

Inducing behavior with abduction

- **Abductive explanation** or **policy** for framework $(\Pi, \varepsilon, G{:}[\alpha,\beta])$
  - $H \subseteq \varepsilon$
  - $\Pi \cup H$ is consistent
  - $\Pi \cup H \models G{:}[\alpha,\beta]$

- Example:

armedAttack: [0.85, 0.95] ← *interOrganizationConflicts*.
kidnap: [0.68, 0.78] ← *unDemocratic* ∧ *internalConflicts*.
bomb: [0, 0.5] ← *externalSupport* ∧ *interOrganizationConflicts*.
bomb: [0.93, 1.0] ← *statusMilitaryWing*(*standing*).

Inducing behavior with abduction

- **Abductive explanation** or **policy** for framework $(\Pi, \varepsilon, G{:}[\alpha,\beta])$
  - $H \subseteq \varepsilon$
  - $\Pi \cup H$ is consistent
  - $\Pi \cup H \models G{:}[\alpha,\beta]$

- Example:

Goal
Ensure probability of bombing is less than 50% *bomb*:[0,0.5]

Abducibles
externalSupport *interOrganizationConflicts*

Inducing behavior with abduction

- **Abductive explanation** or **policy** for framework $(\Pi, \varepsilon, G{:}[\alpha,\beta])$
  - $H \subseteq \varepsilon$
  - $\Pi \cup H$ is consistent
  - $\Pi \cup H \models G{:}[\alpha,\beta]$

- Example:

Goal
Ensure probability of bombing is less than 50% *bomb*:[0,0.5]

Abducibles
externalSupport *interOrganizationConflicts*

- How do we find a solution?

Solving probabilistic logic abduction

- Choose combination of abducibles (possible policy)
- Check if it is a solution
  - Add abducibles to the state
  - Get the applicable rules from the probabilistic logic program
  - Solve an entailment query for the goal
- Done!

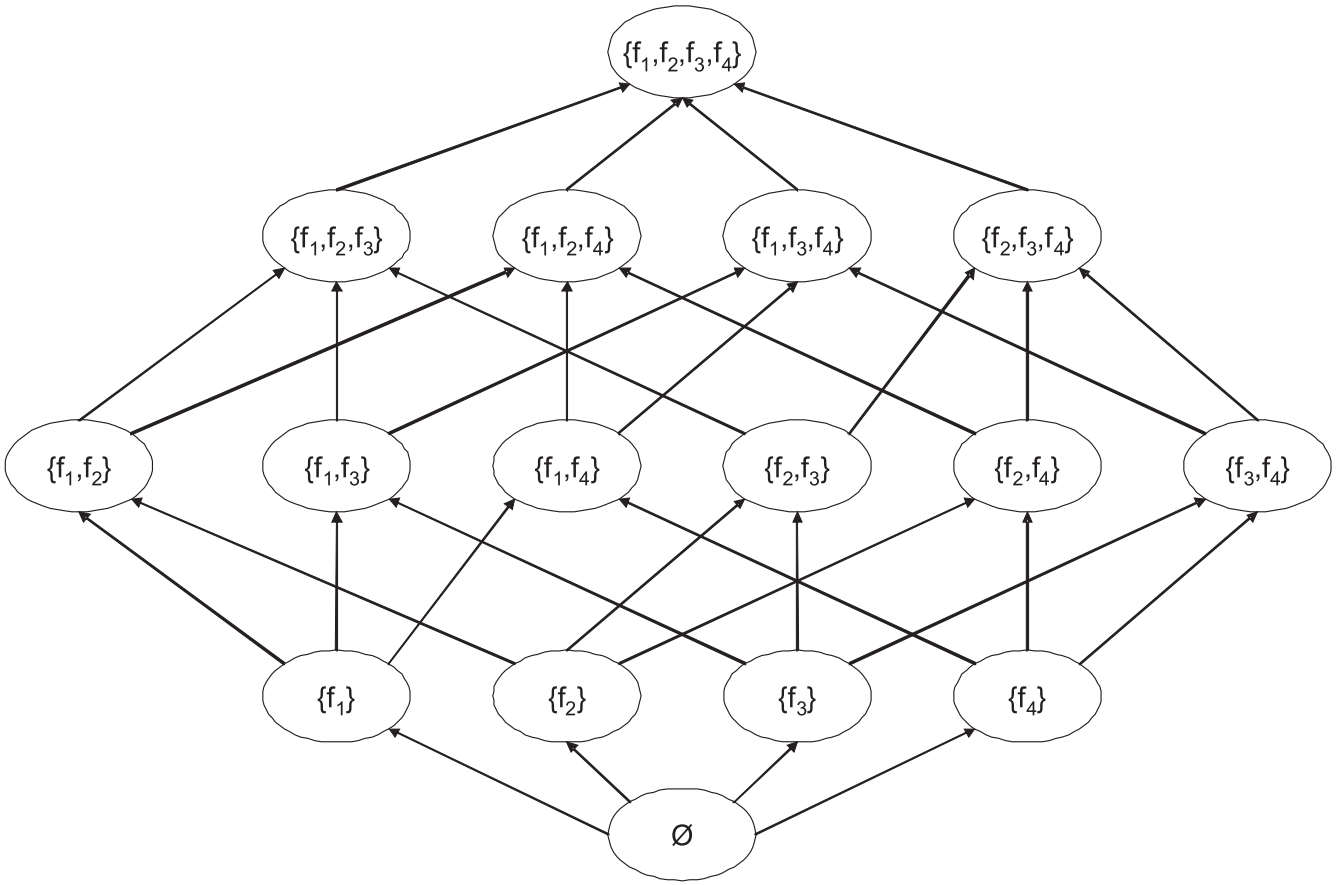
- **Issues?**

Solving probabilistic logic abduction

- Choose combination of abducibles (possible policy)
- Check if it is a solution
  - Add abducibles to the state
  - Get the applicable actions from the probabilistic logic program
  - Solve an entailment query for the goal
- Done!

- Issues?
  - Constraint satisfaction problem can be expensive
  - Have to look at all possible combinations of abducibles?!?
    - What if we want the “best” policy, not just any policy?

Explanations form a lattice



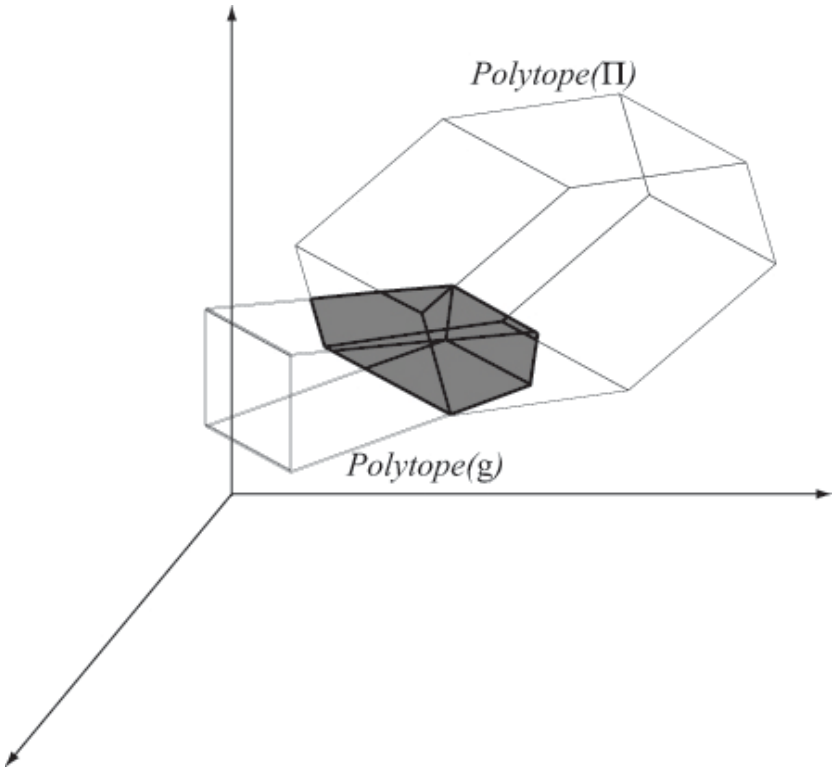
- Naïve algorithm—traverse lattice to find simplest solution
- **Heuristic** approach—use evaluation function for pruning

Geometric interpretation of abduction

- Remember constraints form convex polytope
  - Space of probability distributions over worlds that are possible given
    - Current state
    - Applicable behavioral rules
- Goal defines a polytope, too!
  - Space of probability distributions over worlds that are possible given goal and probability bounds
- Abduction is the problem of finding some new polytope in the **overlapping** region

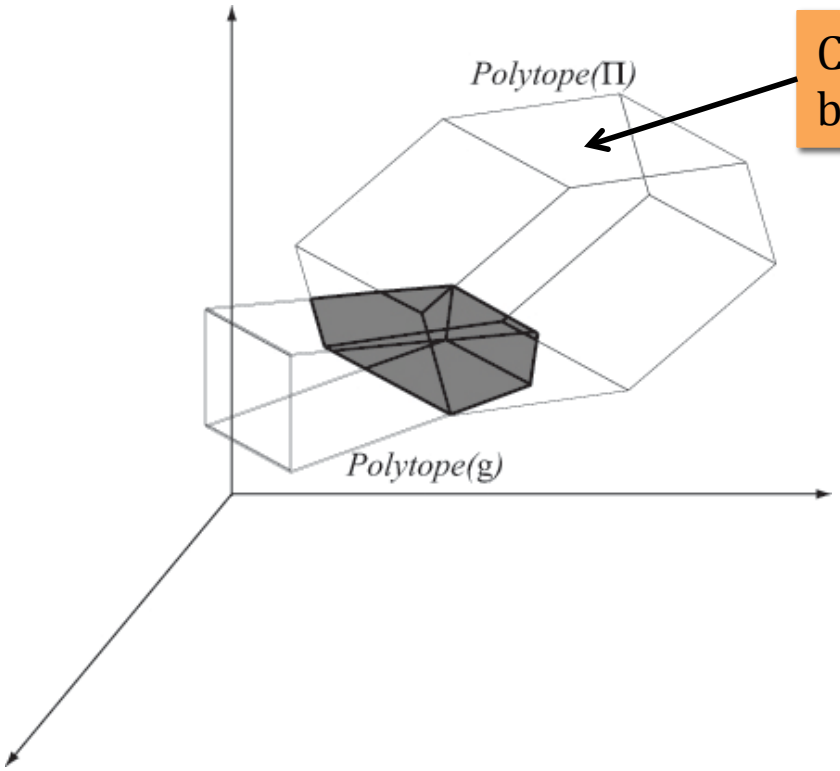
Slicing the current world

- Abduction is essentially adding additional constraints
  - How can we change the current world to constrain behavior to something we want to achieve?



Slicing the current world

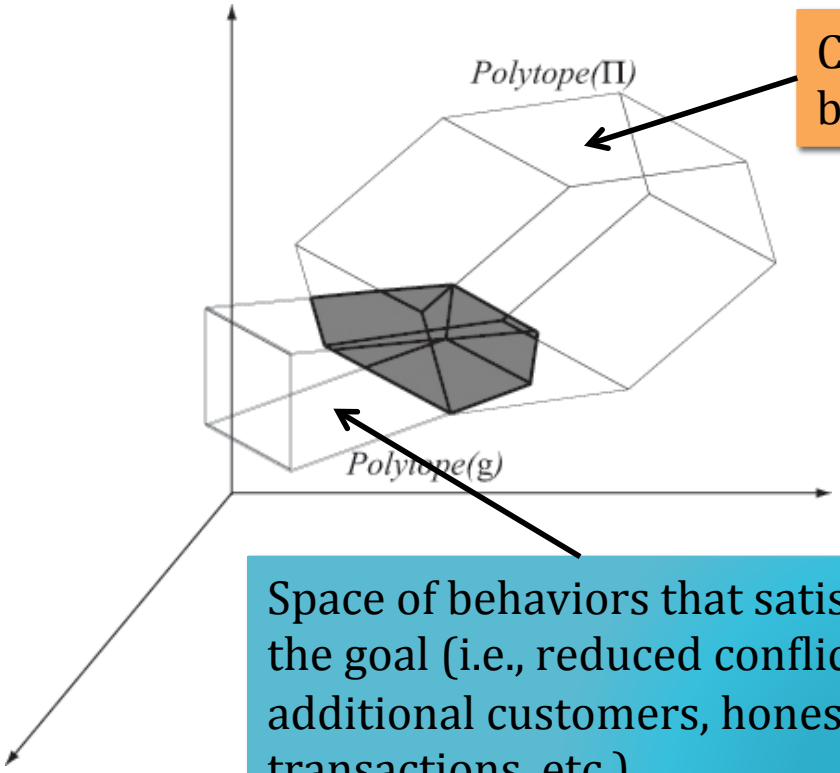
- Abduction is essentially adding additional constraints
  - How can we change the current world to constrain behavior to something we want to achieve?



Current situation—all possible behaviors allowed by our model

Slicing the current world

- Abduction is essentially adding additional constraints
  - How can we change the current world to constrain behavior to something we want to achieve?



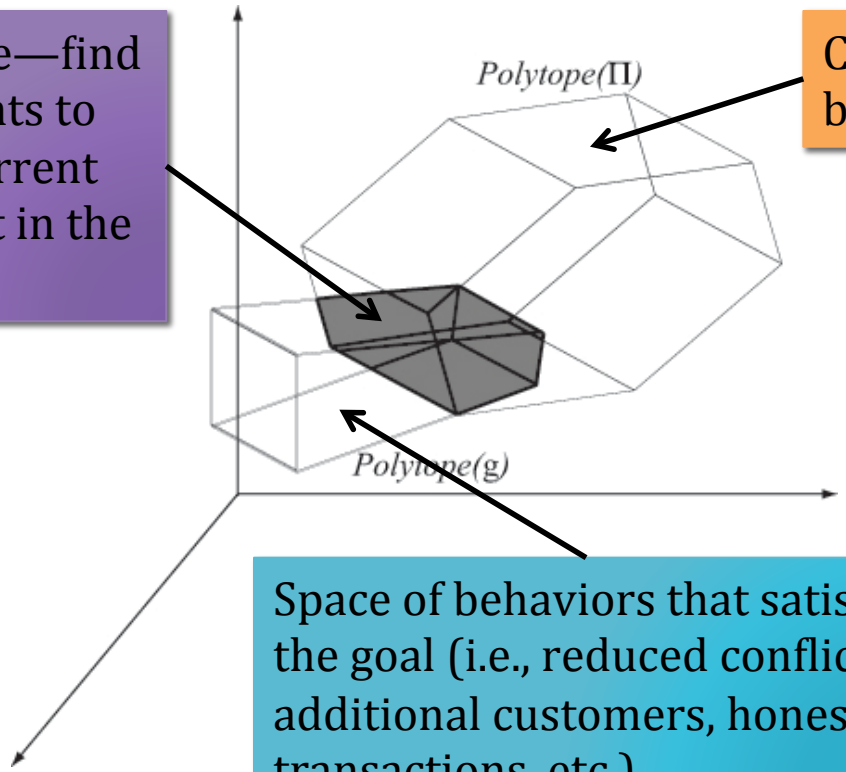
Current situation—all possible behaviors allowed by our model

Space of behaviors that satisfy the goal (i.e., reduced conflict, additional customers, honest transactions, etc.)

Slicing the current world

- Abduction is essentially adding additional constraints
  - How can we change the current world to constrain behavior to something we want to achieve?

Solution space—find new constraints to reduce the current situation to fit in the goal space



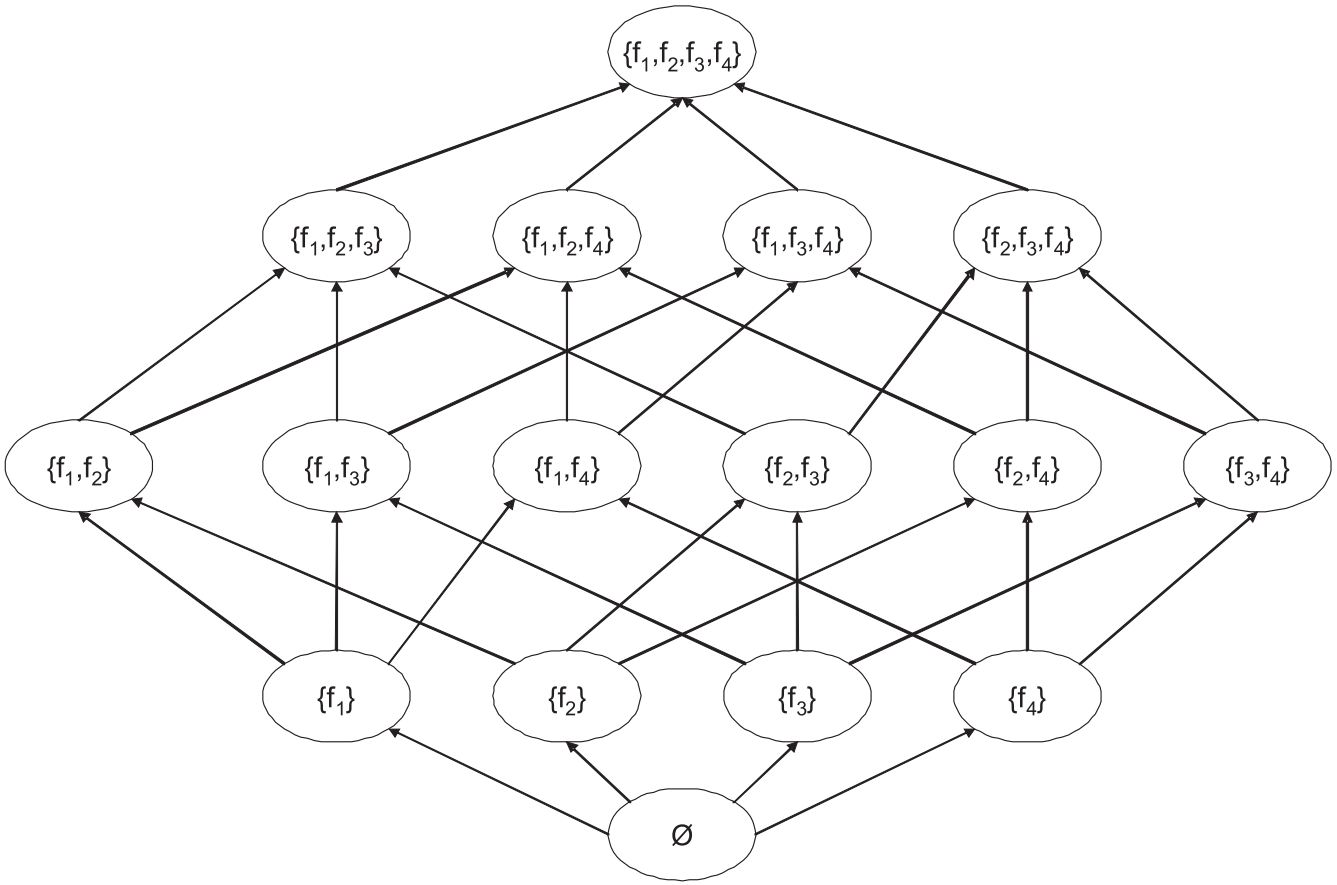
Current situation—all possible behaviors allowed by our model

Space of behaviors that satisfy the goal (i.e., reduced conflict, additional customers, honest transactions, etc.)

Search heuristics informed by geometry

- When can we prune the lattice of solutions?
  1. There is no polytope defined by $\Pi \cup H$, i.e., $\Pi \cup H$ is **inconsistent**
     - Prune **supersets** of H—will also be inconsistent
  2. The polytope defined by $\Pi \cup H$ is not contained in the goal
     - Prune **subsets** of H—not slices that will make the polytope fit in the goal
  3. The polytope defined by $\Pi \cup H$ is not contained in the goal G:[α,β] s.t. $\Pi \cup H$ is contained in the polytope G:[α',β'] s.t. [α',β'] ∩ [α,β] (i.e., the original situation has been sliced too much)
     - Prune **supersets** of H—no additional cuts get us closer to the goal

Design an evaluation function for pruning



- Choosing from the “middle” gives us the most pruning opportunities
- Assign values s.t. “medium” sets have higher value

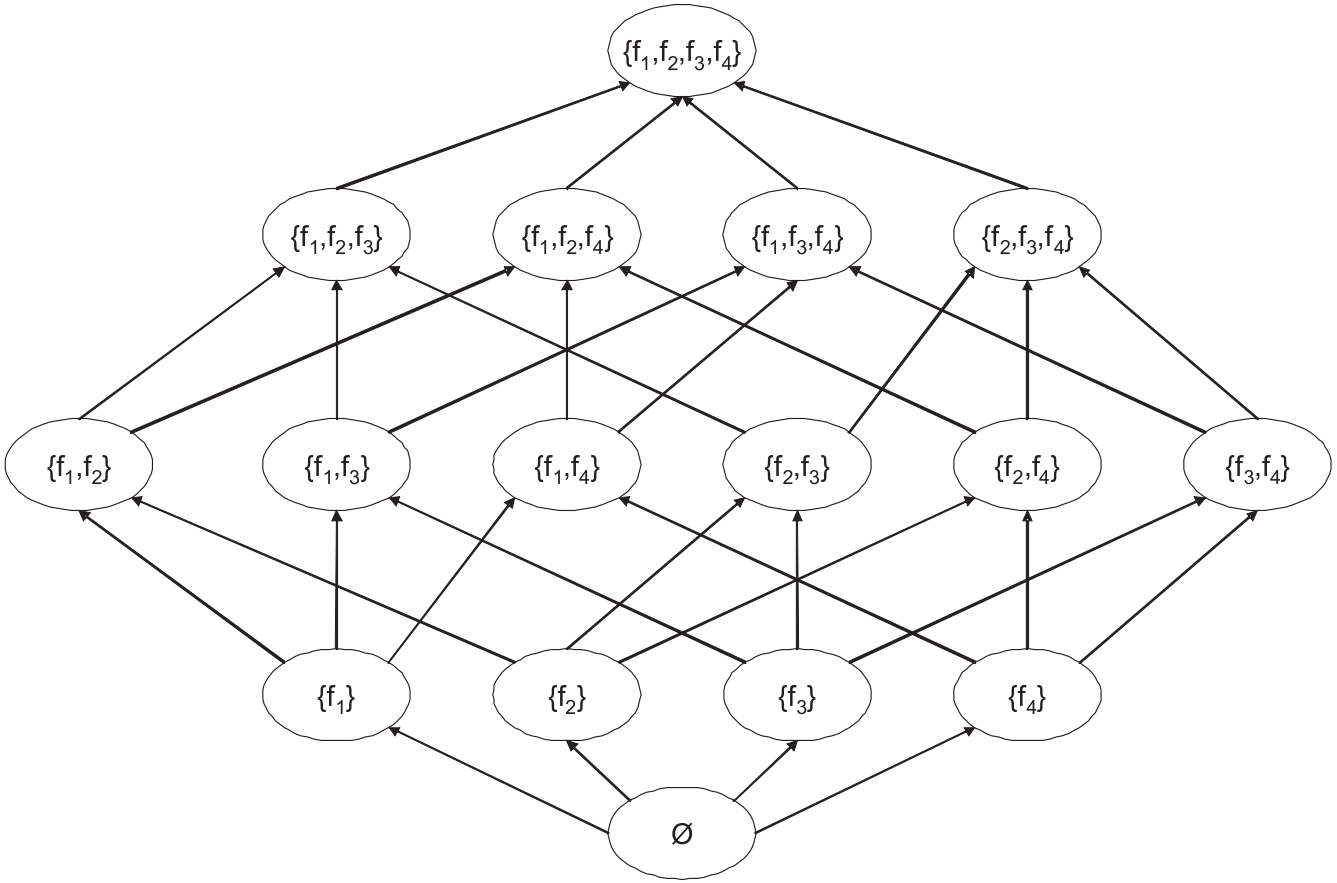
Heuristic for abductive search

- For a possible abductive solution (a policy) H
  - $v_{max} = max\{|H'| \text{ s.t. } H \subseteq H'\}$
    - Size of largest superset
  - $v_{min} = min\{|H'| \text{ s.t. } H' \subseteq H\}$
    - Size of smallest subset

- Pruning heuristic

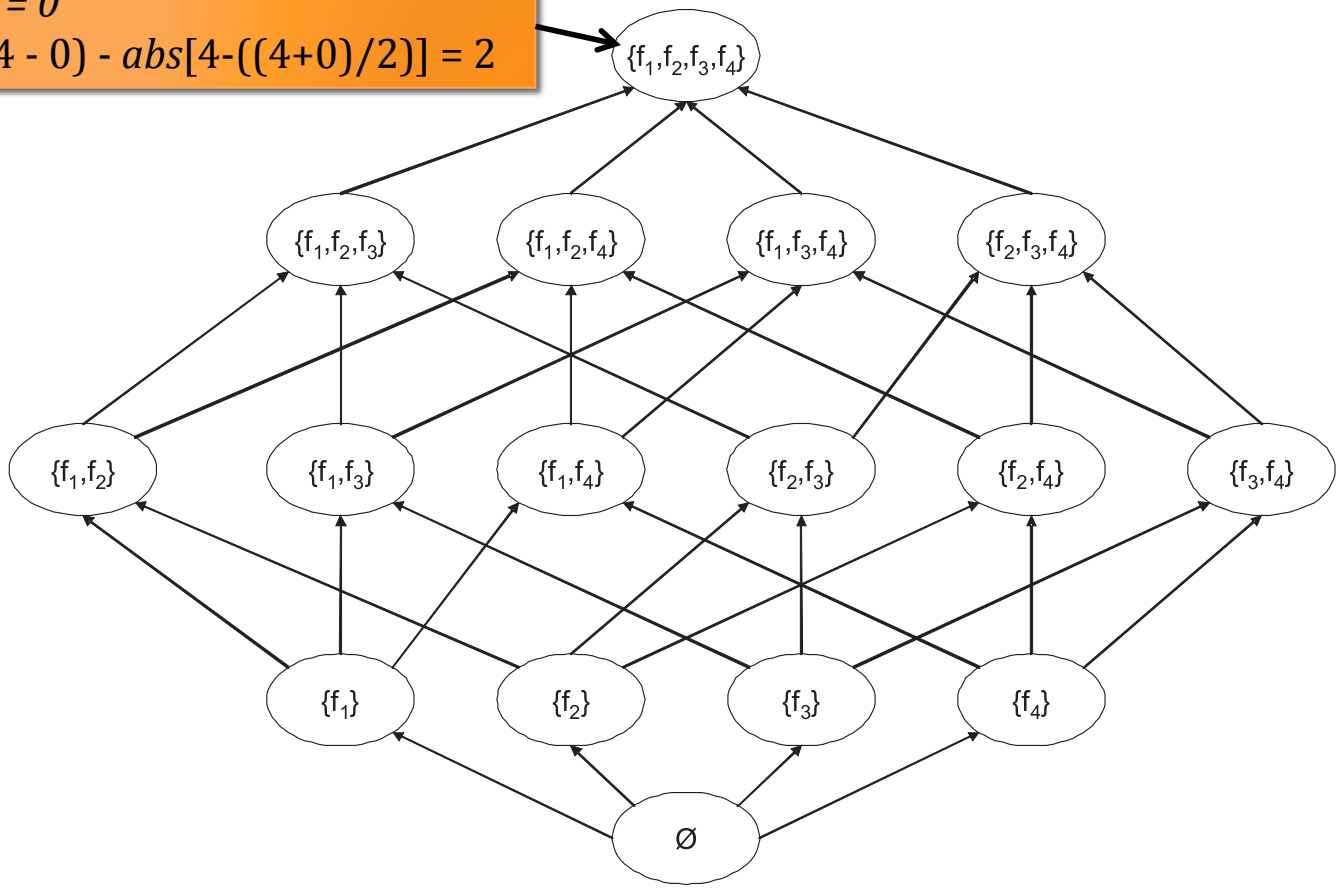
$$score(H) = (v_{max} - v_{min}) - abs\left[|H| - \frac{(v_{max} + v_{min})}{2}\right]$$

Design an evaluation function for pruning



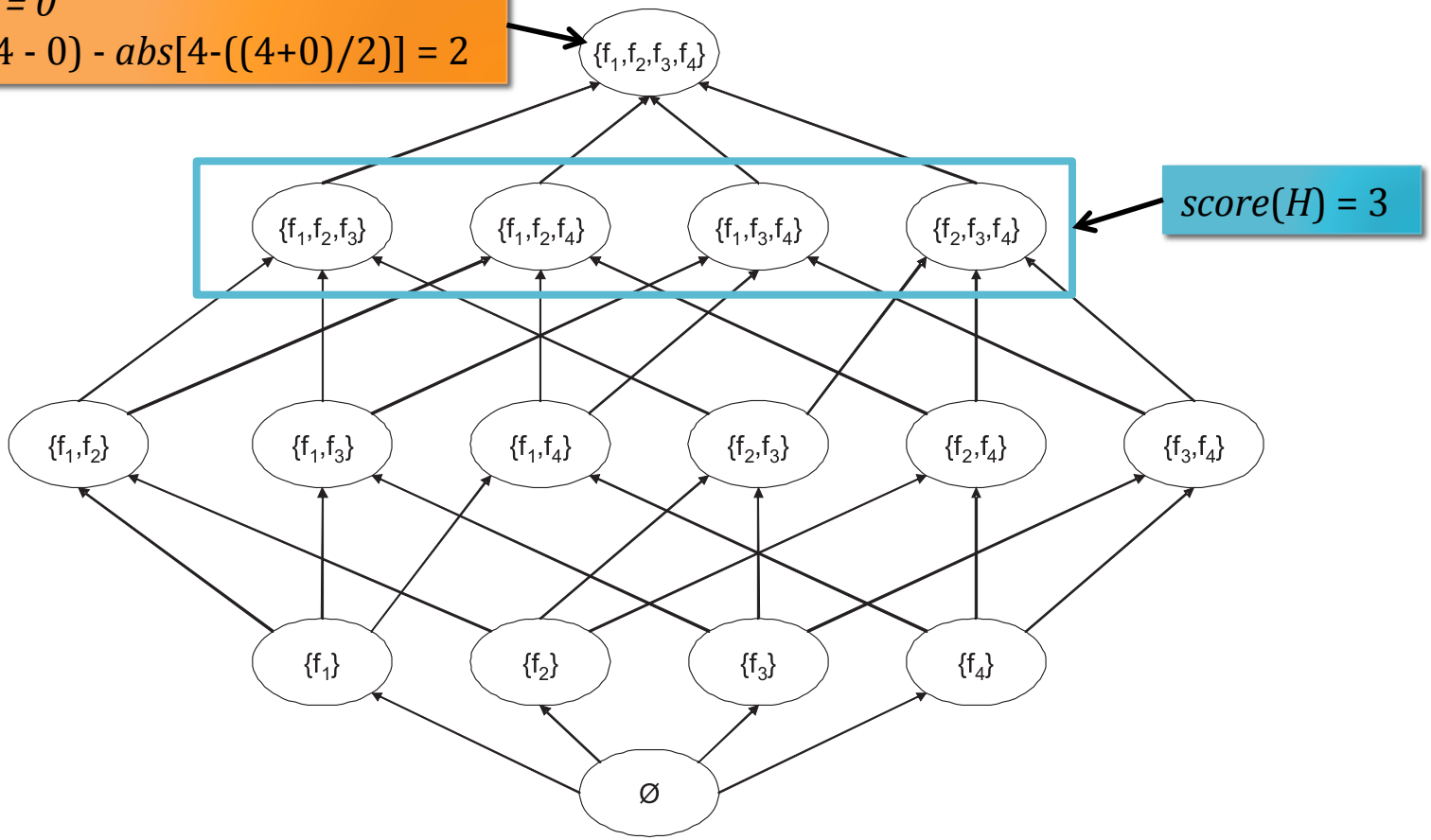
Design an evaluation function for pruning

$v_{max} = 4, v_{min} = 0$
$score(H) = (4 - 0) - abs[4-((4+0)/2)] = 2$



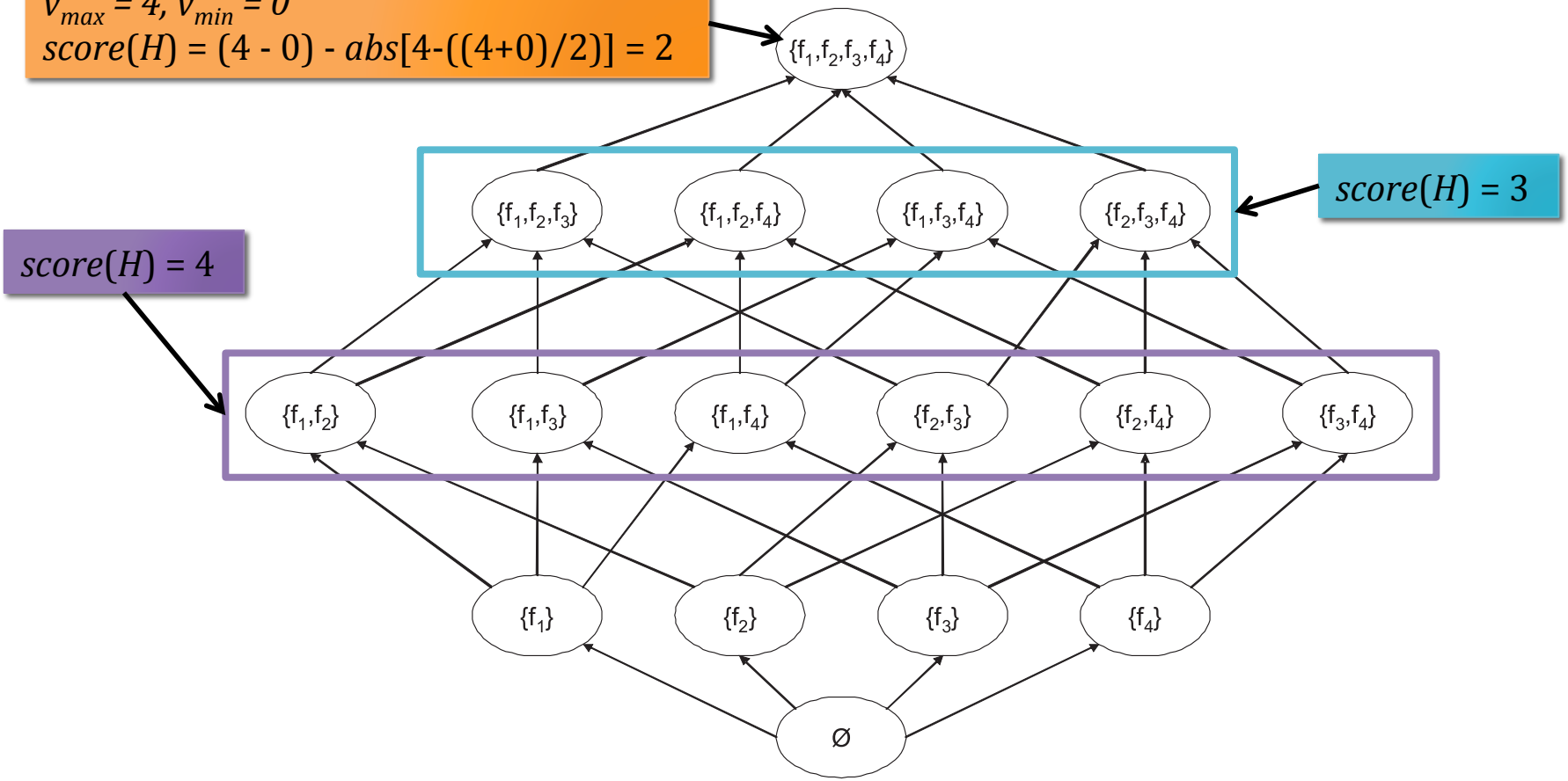
Design an evaluation function for pruning

$v_{max} = 4$, $v_{min} = 0$
$score(H) = (4 - 0) - abs[4-((4+0)/2)] = 2$



Design an evaluation function for pruning

$v_{max} = 4, v_{min} = 0$
$score(H) = (4 - 0) - abs[4-((4+0)/2)] = 2$

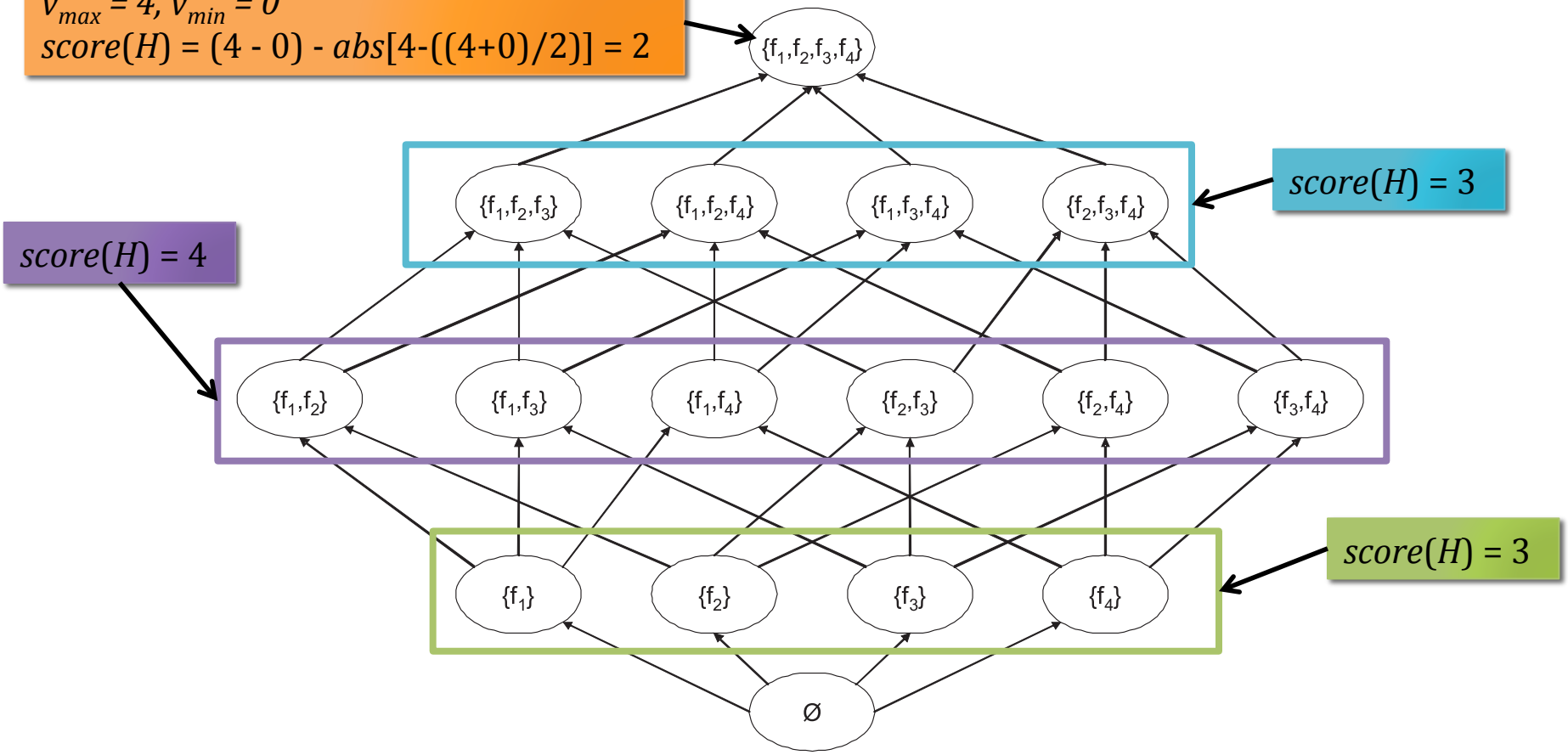
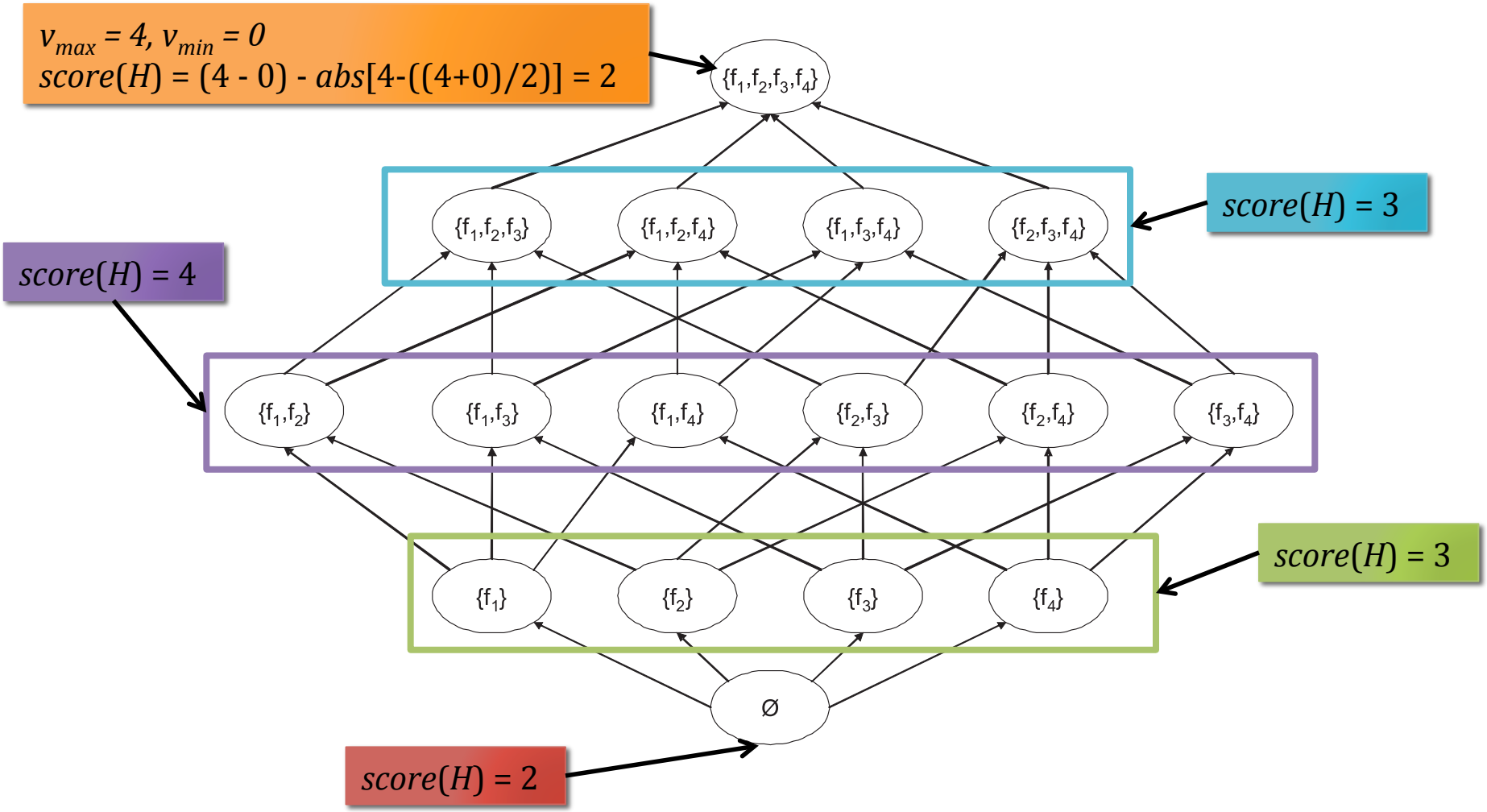


$score(H) = 3$

$score(H) = 4$

Design an evaluation function for pruning

$v_{max} = 4$, $v_{min} = 0$
$score(H) = (4 - 0) - abs[4-((4+0)/2)] = 2$



Design an evaluation function for pruning