

Multi-agent decision making:
preference reasoning and **voting theory**

Outline

- Preferences
- Several kinds of preferences
- Preferences in multi-agent decision making
- Voting theory (social choice)
- In multi-agent AI scenarios:
 - Missing and imprecise preferences
 - Computational concerns
 - Large set of candidates
 - Candidate set with a combinatorial structure

Preferences

- Preferences are **ubiquitous in everyday decision making**
 - Essential ingredients in every reasoning tool
- Preferences are **orderings** over possible options
 - Options: candidates, car, computers, books, movies ...
- Preferences can model **levels of acceptance, or costs**
 - Preferences are tolerant constraints
 - Constraints are strict requirements that must be satisfied

Preferences

- **If all constraints**, possibly
 - no solution, or
 - too many of them, all apparently equally good
- Some problems are **naturally modelled** with preferences
 - I don't like meat, and I prefer fish to cheese
- **Constraints and preferences** may be present in the same problem
 - Ex. Timetabling, ..

University timetable

Professor

I cannot teach on Wednesday afternoon.

I prefer not to teach early in the morning, nor on Friday afternoon.

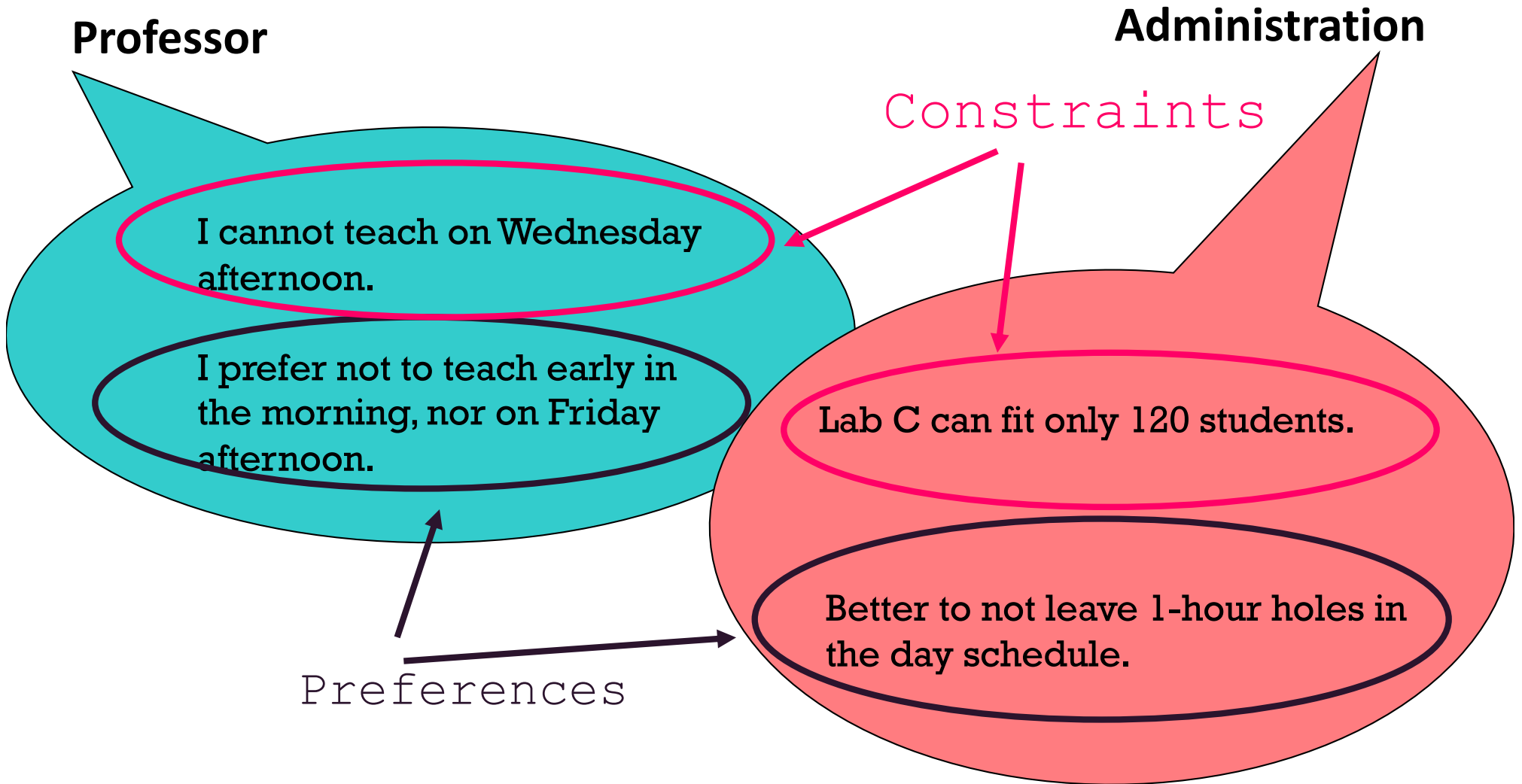
Administration

Constraints

Lab C can fit only 120 students.

Better to not leave 1-hour holes in the day schedule.

Preferences



Several kinds of preferences

■ Unconditional

- *I prefer taking the bus*

■ Conditional

- *I prefer taking the bus if it's raining*

■ Multi-agent

- *I like blue, my husband likes green,
what color for the new car?*

Several kinds of preferences

■ Quantitative

- Numbers, or ordered set of objects

- *My preference for ice cream is 0.8, and for cake is 0.6*

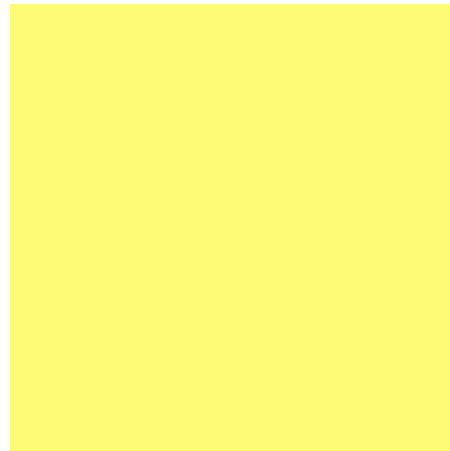
■ Qualitative

- Pairwise comparisons

- *Ice cream is better than cake*

Two main ways to model compactly preferences

- **Several kinds of preferences**
- **Two compact ways to model preferences**
 - **Soft constraints**
for modeling **quantitative** and **unconditional** preferences
 - *Ex., My preference for ice cream is 0.8, and for cake is 0.6*
 - **CP-nets**
for modeling **qualitative** and **conditional** preferences
 - *Ex., Red wine is better than white wine if there is meat*



Multi-agent decision making:
preference reasoning and **voting theory**

II PART

Outline

- Preferences
- Several kinds of preferences
- Preferences in multi-agent decision making
- Voting theory (social choice)
- In multi-agent AI scenarios:
 - Missing and imprecise preferences
 - Computational concerns
 - Large set of candidates
 - Candidate set with a combinatorial structure

Preferences for collective decision making in multi-agent systems

- Several **agents**
- **Common** set of **possible decisions**
- **Each agent** has its preferences over the possible **decisions**
- **Goal:** to **choose one** of the **decisions**, **based on the preferences** of the agents
 - or a set of decisions
 - or a ranking over the decisions
- **AI scenarios** add:
 - Imprecision
 - Uncertainty
 - Complexity concerns
 - Combinatorial structure of the decisions

Applications

■ Doodle



- **Several time slots** under consideration
- Participants **accept** or **reject** each time slot
 - Very simple way to **express preferences** over time slots
 - Very little information communicated to the system
- **Collective choice**: a single time slot
 - The one with most acceptance votes from participants

■ Other applications

- Group recommender systems
- Meta-search engine

How to compute a collective decision?

- Let the **agents** **vote** by expressing their **preferences** over the **possible decisions**
- **Aggregate** the **votes** to get a **single decision**
- Let's look at **voting theory**
 - Agents = Voters
 - Decisions = Candidates
 - Preferences
 - Chosen decision = winner

Voting theory (Social choice)

- **Voters**
- **Candidates**
- **Each voter** expresses its **preferences** over the **candidates**
- **Goal:** to choose one **candidate** (the winner), based on the voters' preferences
 - Also many candidates, or ranking over candidates
- **Voting Rules** (functions) to achieve the goal



Some voting rules

■ Plurality

■ **Voting:** each voter provides **the most preferred decision**

■ **Selection:** the **decision** preferred by the **largest** number of voters

■ **Majority:** like **plurality**, over **2** options

Vote for one option.

☐

Joe Smith

☒

John Citizen

☐

Jane Doe

☐

Fred Rubble

☐

Mary Hill

Plurality

- **Voting:** the most preferred decision
- **Selection:** the decision preferred by the largest number of agents

- **Example:**

- 6 voters

- 3 candidates:



Profile



Voter 1

Voter 2

Voter 3

Voter 4

Voter 5

Voter 6

Winner



Another voting rule

- **Approval** (m options)
 - **Voting:**
each voter **approves** any **number of options**
 - **Selection:** **option** with **most votes**

Voting rule used in Doodle

Another voting rule

■ Borda

- **Voting:** each voter provides a ranking over all options
- **Score of an option** for a voter: number of options that it dominates
- **Selection:** option with greatest sum of scores

Borda

rank

3



2



1



0



Voter 1

rank

3



2



1



0



Voter 2

rank

3



2



1



0



Voter 3

rank

3



2



1



0



Voter 4

rank

3



2



1



0



Voter 5

Borda
Count

9

8

7

6

Winner



Some desirable properties

■ Unanimity (efficiency)

- If all voters have the **same top choice**, it is selected

■ Non-dictatorship

- There is **no dictator**
- **Dictator: voter** such that **his top choice always wins**, regardless of the votes of other voters

■ Non-manipulability

- There is **no incentive** for agents to misrepresent the preferences

Two classical impossibility results

■ Arrow's theorem (1951)

- **Totally ordered** preferences
- It is **impossible** to find a **voting rule** with some desirable properties including
 - **unanimity**
 - **non-dictatoriality**



Nobel prize
in Economics 1972



■ Gibbard-Satterthwaite's theorem (1973)

- **Totally ordered** preferences
- it is **impossible** to have a reasonable **voting rule** that is
 - **non-dictatorial**
 - **non-manipulable**



- These impossibility results **hold also** when we allow **partially ordered preferences**



Multi-agent decision making:
preference reasoning and **voting theory**

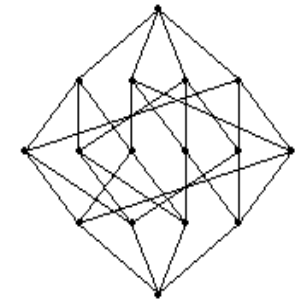
III PART

Voting theory and multi-agent systems

- Voting theory (social choice)
 - Voting rules
 - Desirable properties
 - Impossibility results
- In multi-agent AI scenarios, we usually have
 - Incomparability
 - Uncertainty, vagueness, preference elicitation
 - Computational concerns
 - Large set of decisions (candidates) w.r.t. number of agents (voters)
 - Combinatorial structure for the set of decisions (candidates)

Incomparability

- **Preferences** do **not always** induce a **total order** over the options
- Preferences may induce a **partial order** where some options are incomparable
- **Some options** are naturally **incomparable**
 - Eg., it may be **easy** to compare two apartments
 - but it may be **difficult** to compare an **apartment** and a **house**, thus we say they are incomparable
- An agent may have several possibly **conflicting preference criteria** he wants to follow
 - Eg., a **cheap and slow** car is incomparable w.r.t. an **expensive and fast** car
- Many **AI formalisms** to model preferences **allow for partial orders** (eg., soft constraints)



Uncertainty, vagueness

■ Missing preferences

- Too costly to compute them
- Privacy concerns
- Ongoing elicitation process



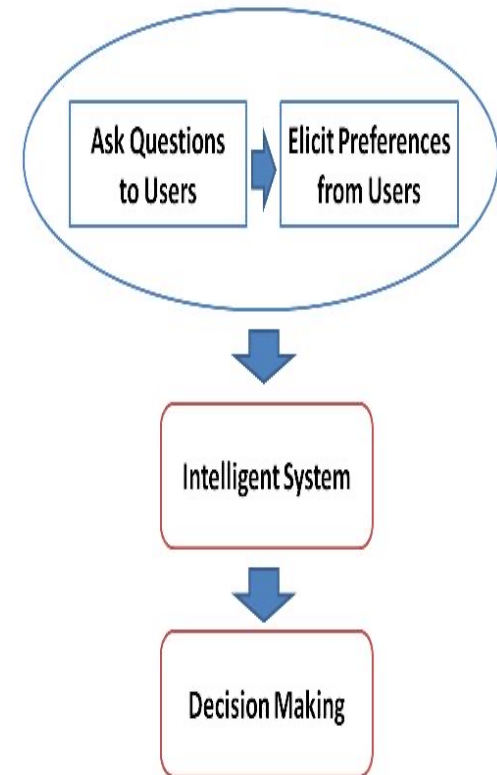
■ Imprecise preferences

- Preferences coming from sensor data
- Too costly to compute the exact preference
- Estimates

Aim in AI: Find compact preference formalisms and solving techniques to model and solve problems with missing or imprecise preferences

Preference elicitation

- Some **preferences** may be **missing**
- Time consuming, **costly**, difficult, to **elicit all preferences**
- We want to **terminate preferences elicitation** as soon as a winner fixed



Computational concerns

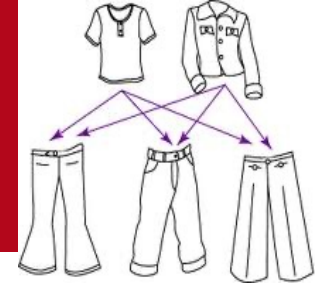
- We would like to avoid very costly ways to
 - Model agents' preferences
 - Compute the winner
 - Reason with the agents' preferences
- On the other hand, we need a computational barrier against bad behaviors (such as manipulation)

Bartholdi, Tovey, Trick. The computational difficulty of manipulating an election. Social Choice and Welfare 1989

Large set of candidates

- In **AI scenarios**, usually
the **set of decisions** (candidates) is **much larger** than
the **set of agents** expressing preferences
over the decisions
- *Eg., many web pages, few search engines*

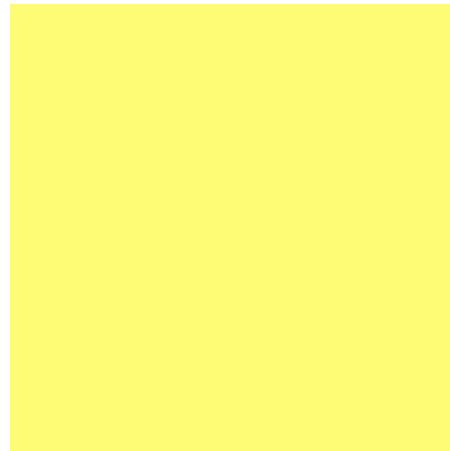
Combinatorial structure for the set of decisions



- The **set of decisions** may have a **combinatorial structure**
- **Dinner example:**
 - Three friends need to decide **what to cook for dinner**
 - 4 items (pasta, main, dessert, drink)
 - 5 options for each item
 - $5^4 = 625$ **possible dinners**
 - It is **unfeasible** providing a **preference ordering over 625 dinners**
- In general: Cartesian product of several variable domains
 - **Variables** = **items** of the menu
 - **Domain** of each variable = **5 options**
- **Goal:** Find **compact preference formalism** to express **agents' preferences**

Formalisms to model preferences compactly

- **Preference ordering over a large set of decisions**
 - ➔ need to **model them compactly**
 - Otherwise too much space and time to handle such preferences
- Two examples:
 - Soft constraint formalism
 - CP-net formalism



Multi-agent decision making:
preference reasoning and **voting theory**

IV PART

Soft constraint formalism

- **Soft constraint formalism** (the c-semiring framework)
 - The agent expresses his preferences over partial assignments of the decision variables
 - From these preferences → the preference ordering over the solution space is generated
 - Any ordering can be obtained!

Formalisms to model compactly preferences

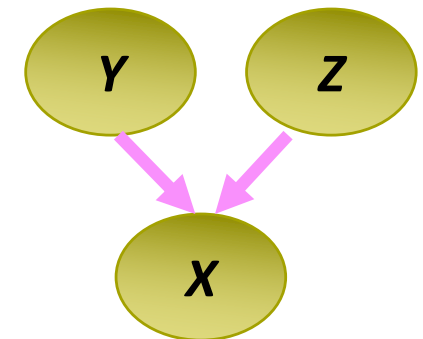
- **Soft constraints** model **quantitative** and **unconditional preferences**
 - Quantitative → a level of preference for each assignment of the variables in a soft constraint
 - It is **difficult to elicitate quantitative** preferences from user
- Moreover, many problems **need statements** like
 - “I like white wine **if** there is fish” (**conditional**)
 - “white wine is **better than** red wine” (**qualitative**)



- **CP-net:** formalism to compactly represent **qualitative** and **conditional preferences**

CP-net (conditional preference network)

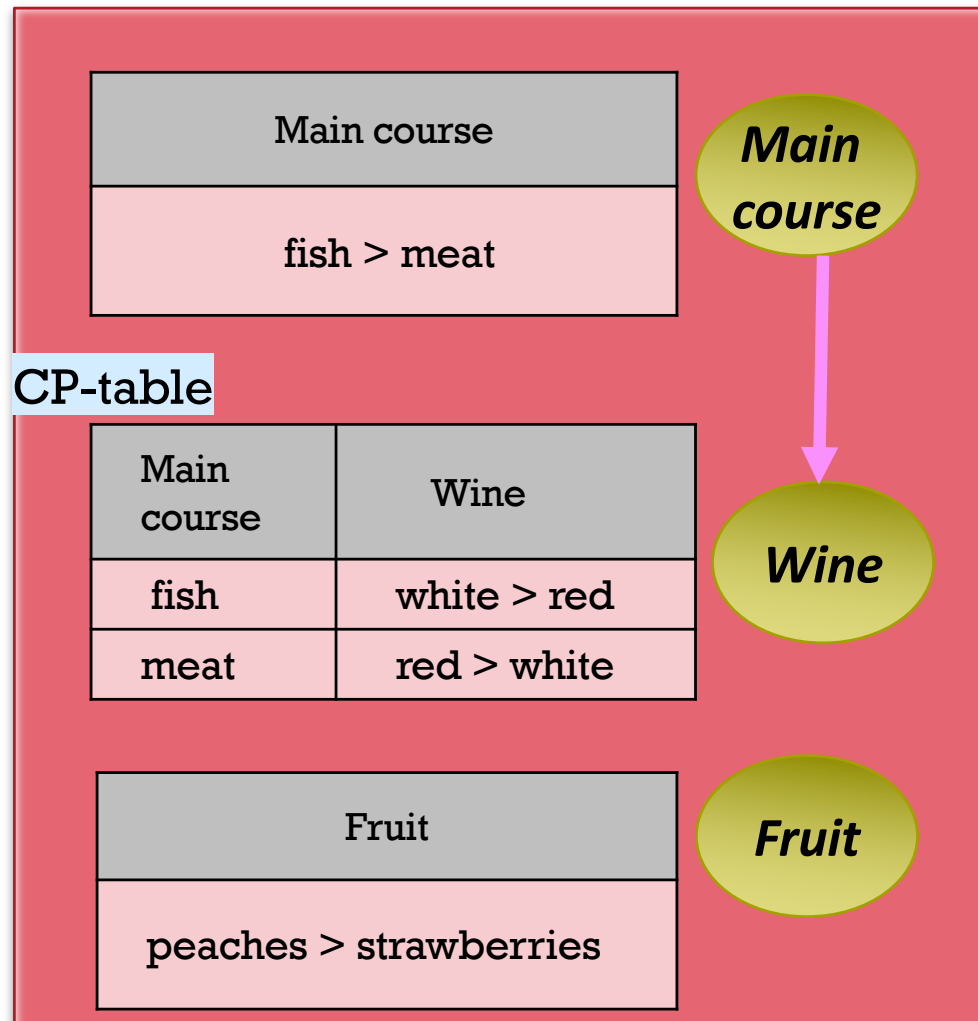
- Variables $\{X_1, \dots, X_n\}$ with domains
- For each variable, a **total order** over its values
- Independent variable:
 - $X=v_1 > X=v_2 > \dots > X=v_k$
- Dependent variable: a total order over its values for each combination of values of some other variables
 - $Y=a, Z=b: X=v_1 > X=v_2 > \dots > X=v_k$
 - X depends on Y and Z (parents of X)
- Graphically: **directed graph** over X_1, \dots, X_n
- Possibly cyclic



CP-net

- A **CP-net** over variables $V = \{X_1, \dots, X_n\}$ is
 - a **directed graph G** over X_1, \dots, X_n
 - whose nodes are annotated with **conditional preference tables** $CPT(X_i)$ for each $X_i \in V$.
 - Each conditional preference table $CPT(X_i)$ associates a **total order \succ** with **each instantiation u of X_i 's parents $Pa(X_i) = U$** .

CP-net: an example



Variables

- **MainCourse**
- **Wine**
- **Fruit**

Domains

- $D_{\text{MainCourse}} = \{\text{meat}, \text{fish}\}$
- $D_{\text{Wine}} = \{\text{white}, \text{red}\}$
- $D_{\text{Fruit}} = \{\text{peaches}, \text{strawberries}\}$

Independent variables

- **MainCourse**
- **Fruit**

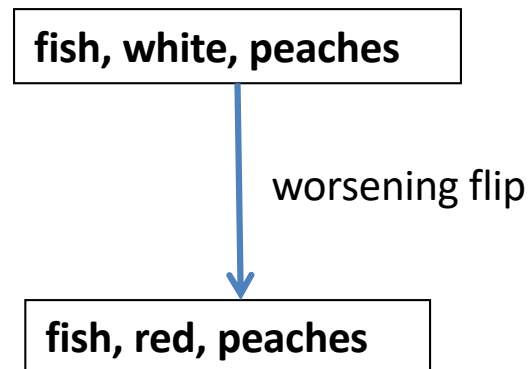
Dependent variable

- **Wine**

CP-net semantics

- **Worsening flip:** changing the value of a variable in a way that is less preferred in some statement

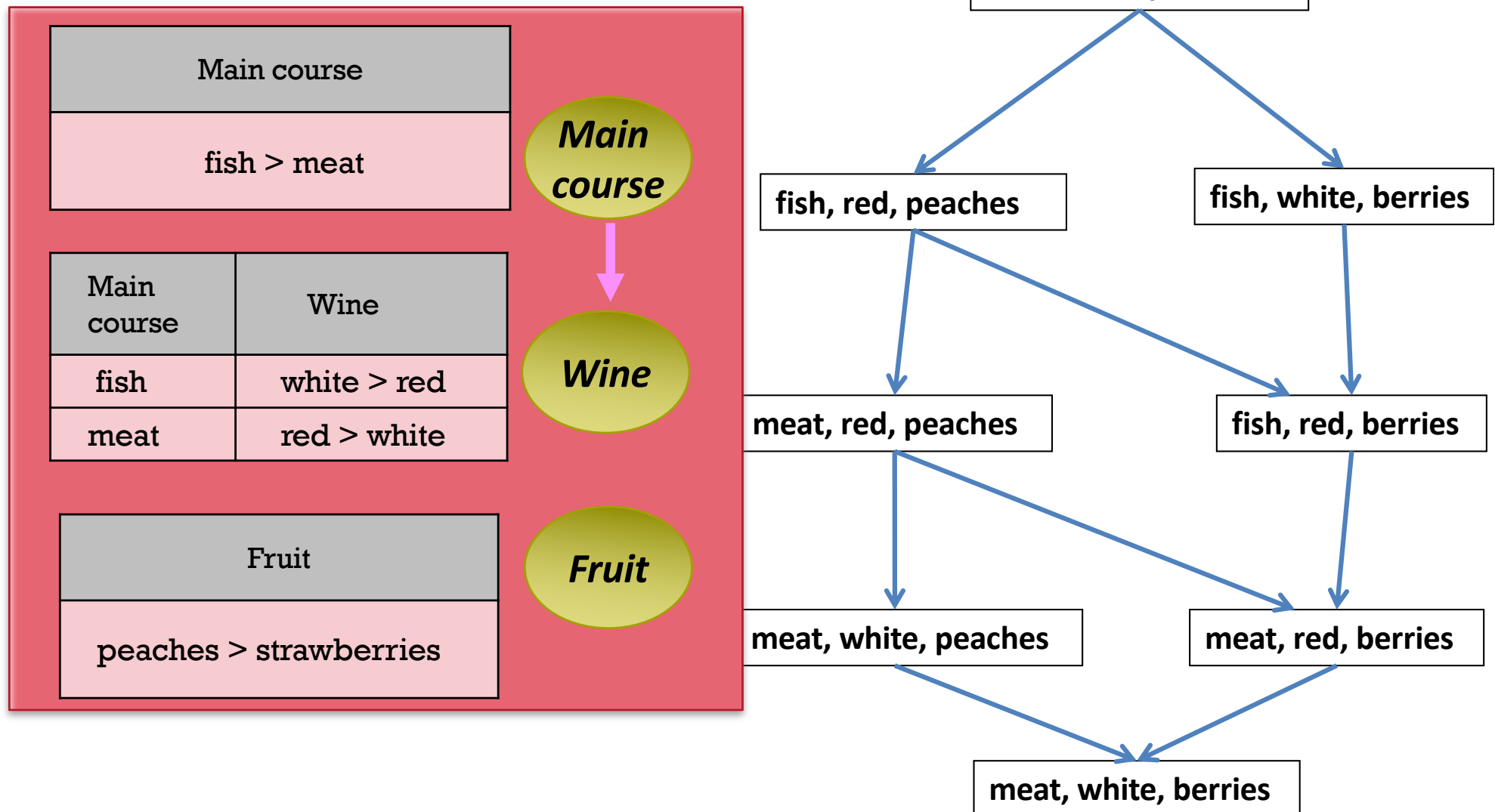
- **Example:**



- An outcome O_1 is **preferred** to O_2 iff there is a sequence of worsening flips from O_1 to O_2
- **Optimal outcome:** if no other outcome is preferred

A CP-net induces an ordering over solutions

Optimal solution

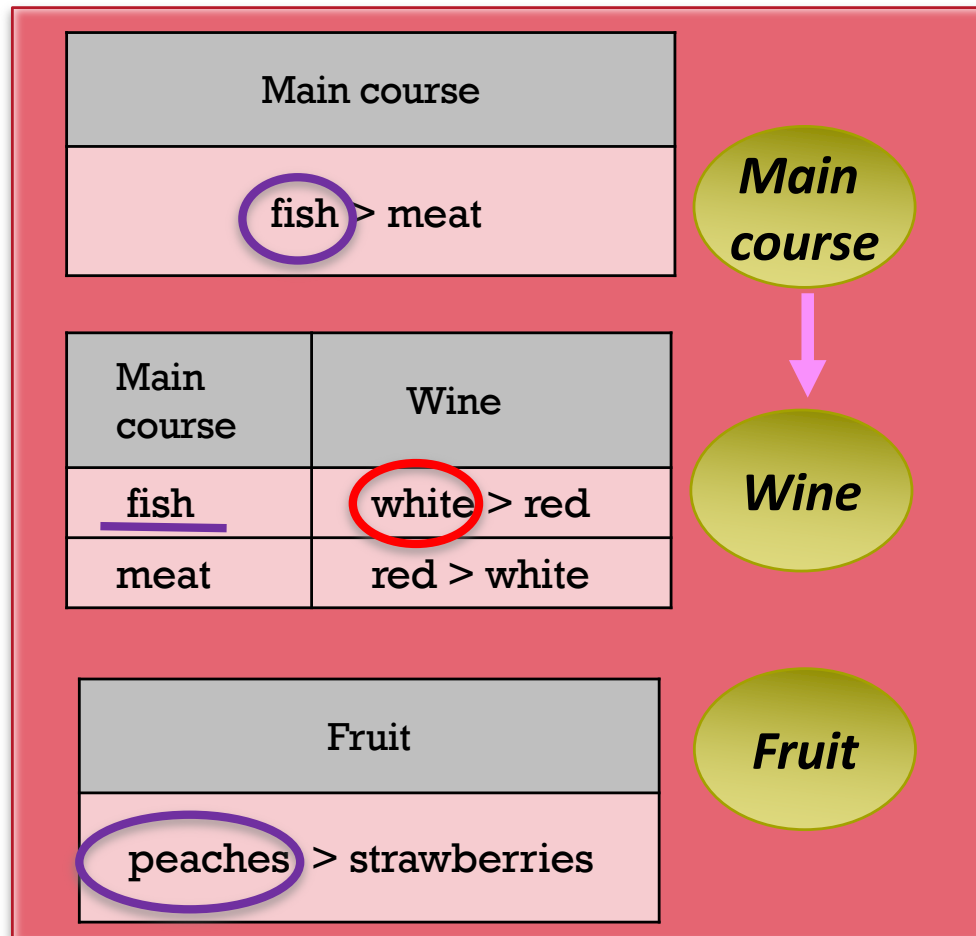


Finding an optimal solution in acyclic CP-nets is easy!

Forward sweep algorithm

- First consider **independent variables**
Assign them their **most preferred values**
- Then consider **dependent variables**, that directly depend on the assigned variables
Assign them their **most preferred values** that are **consistent** with the **values previously assigned** to their parents
- And so on until we assign a value to all the variables

Finding an optimal solution in acyclic CP-nets is easy!



Optimal solution

fish, white, peaches

Soft constraints vs. CP-nets

- **Different expressive power**
- **Different computational complexity** for reasoning with them

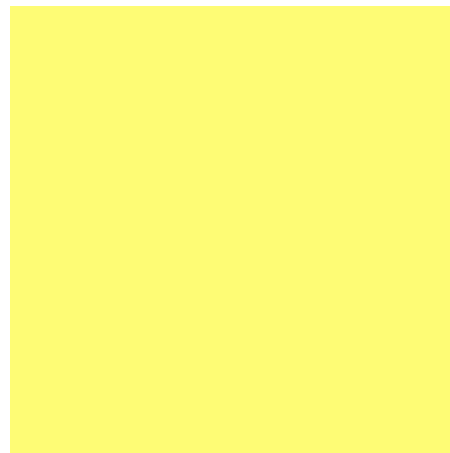
Preference orderings

Find an optimal decision

Compare two decisions

Check if a decision is optimal

Soft CSPs	Tree-like soft CSPs	CP-nets	Acyclic CP-nets
all	all	some	some
difficult	easy	difficult	easy
easy	easy	difficult	difficult
difficult	easy	easy	easy



Multi-agent decision making:
preference reasoning and **voting theory**

V PART

Sequential voting

- **Several voters**
- **Decisions** are made **by several issues**
- Main idea: **vote separately on each issue**, but do so **sequentially**
- This gives **voters** the opportunity to make **their vote for one issue depending** on the **decisions on previous issues**
- In AI: **sequential voting** when agents express their preferences
 - via **soft constraints**
 - via **CP-nets**

Sequential voting with soft constraints

- How to apply **sequential voting** when **agents** express their **preferences** via **soft constraint problems**?
- Assume the agents have
 - the **same constraint graph**
 - but **different preference values**

Dinner example, three agents, fuzzy constraints

Pesto 1
Tom 0.7



Beer 1
Wine 0.7

Agent 1

(Pesto, Beer) 1
(Pesto, Wine) 0.5
(Tom, Beer) 0.7
(Tom, Wine) 0.3

Pesto 0.9
Tom 1



Beer 1
Wine 1

Agent 2

(Pesto, Beer) 1
(Pesto, Wine) 0.9
(Tom, Beer) 0.9
(Tom, Wine) 0.9

Pesto 1
Tom 0.3



Beer 1
Wine 1

Agent 3

(Pesto, Beer) 1
(Pesto, Wine) 0.3
(Tom, Beer) 0.3
(Tom, Wine) 1

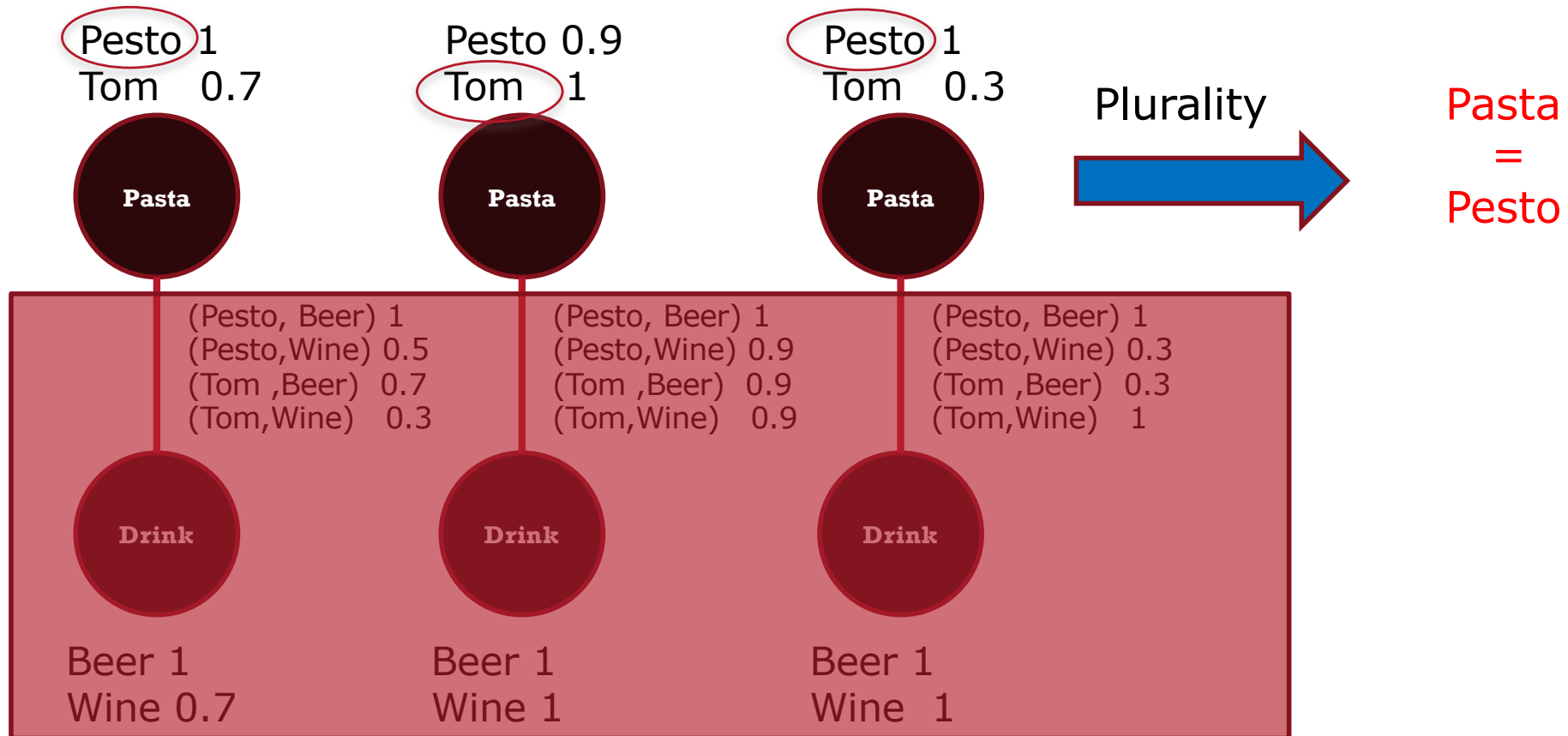
The sequential voting approach

■ For each variable

1. **Compute** an **explicit profile** over the **variable domain**
2. **Apply a voting rule** to this explicit profile →
the rule will return a **specific value**
that will be assigned **to this variable**
3. **Add the information** about the selected variable value

Similar approach used for CP-nets in [Lang, Xia, 2009]

Dinner example using plurality

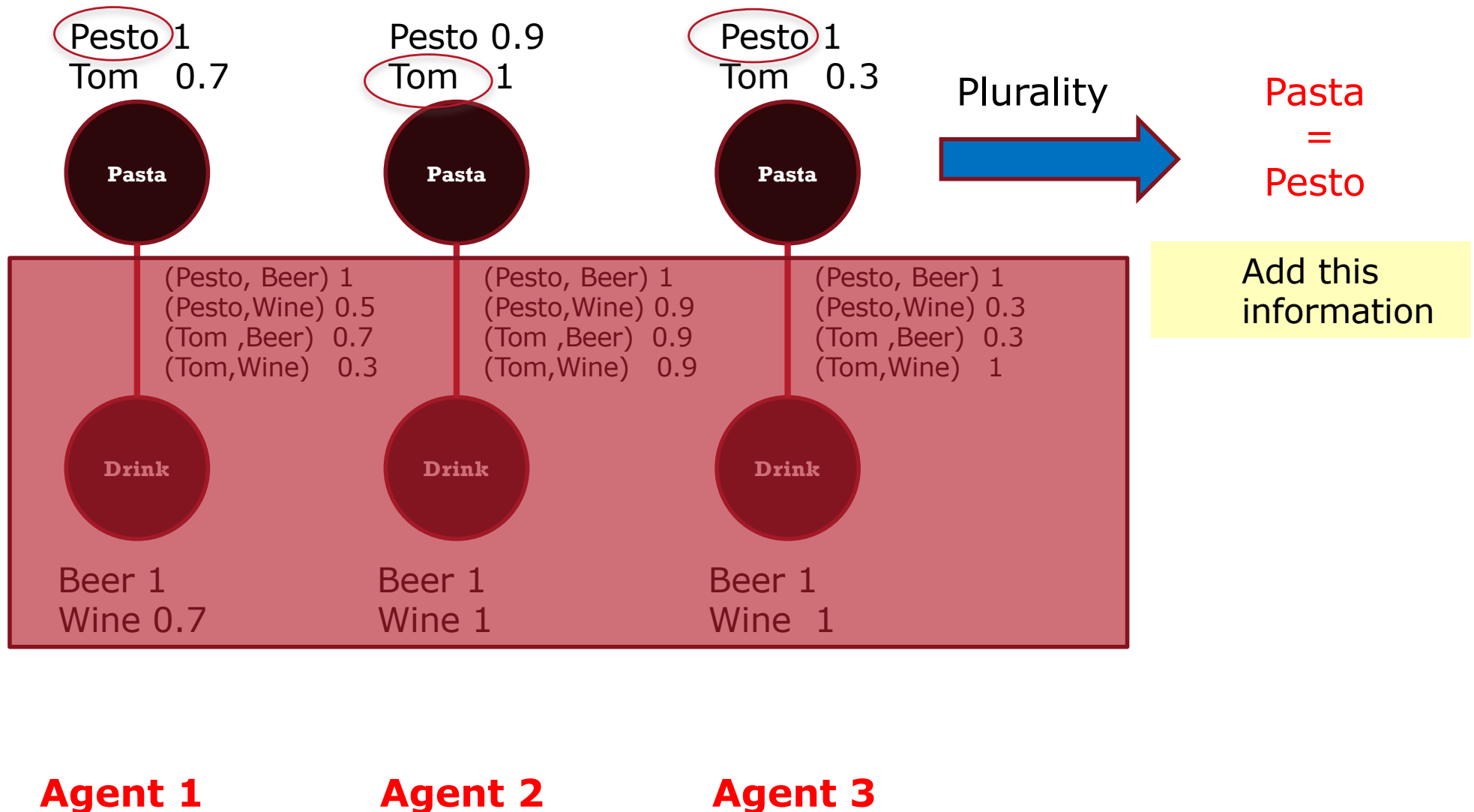


Agent 1

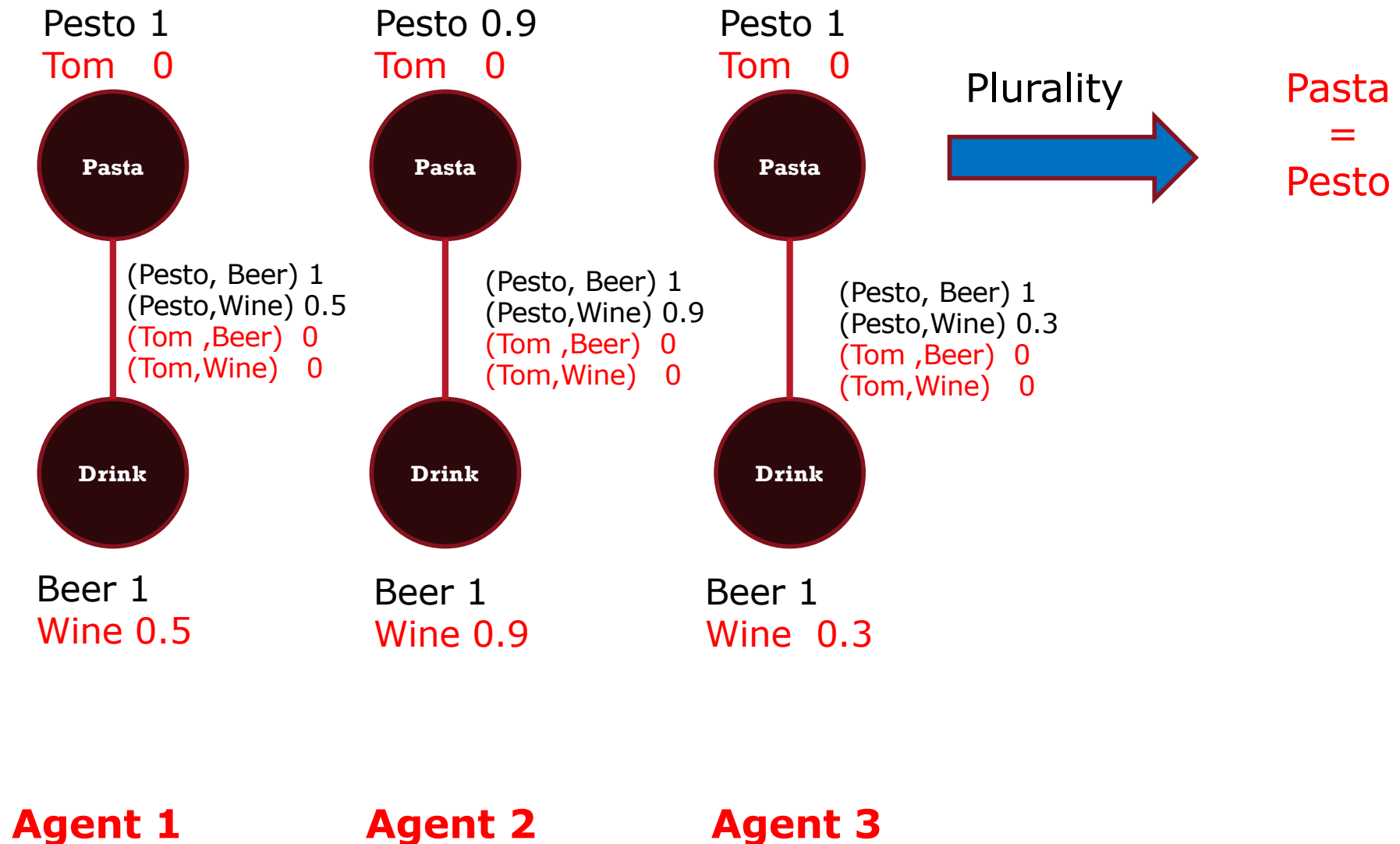
Agent 2

Agent 3

Dinner example using plurality



Dinner example using plurality



Dinner example using plurality

