L5. Reasoning Logically

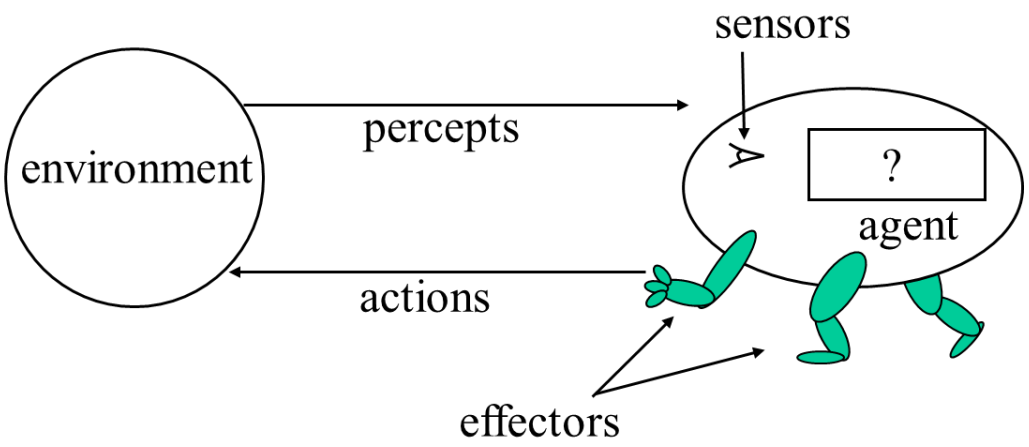
- Knowledge Representation
- Logic and Inference
- Propositional Logic
- First-order logic
- Vumpus World

Assume that
We design an intelligent agent (travel agent, driving agent, …)

What is an intelligent agent?
How can an intelligent agent logically think and act?

What is an agent *A*?

- An **agent** is anything that can viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**.



Examples:

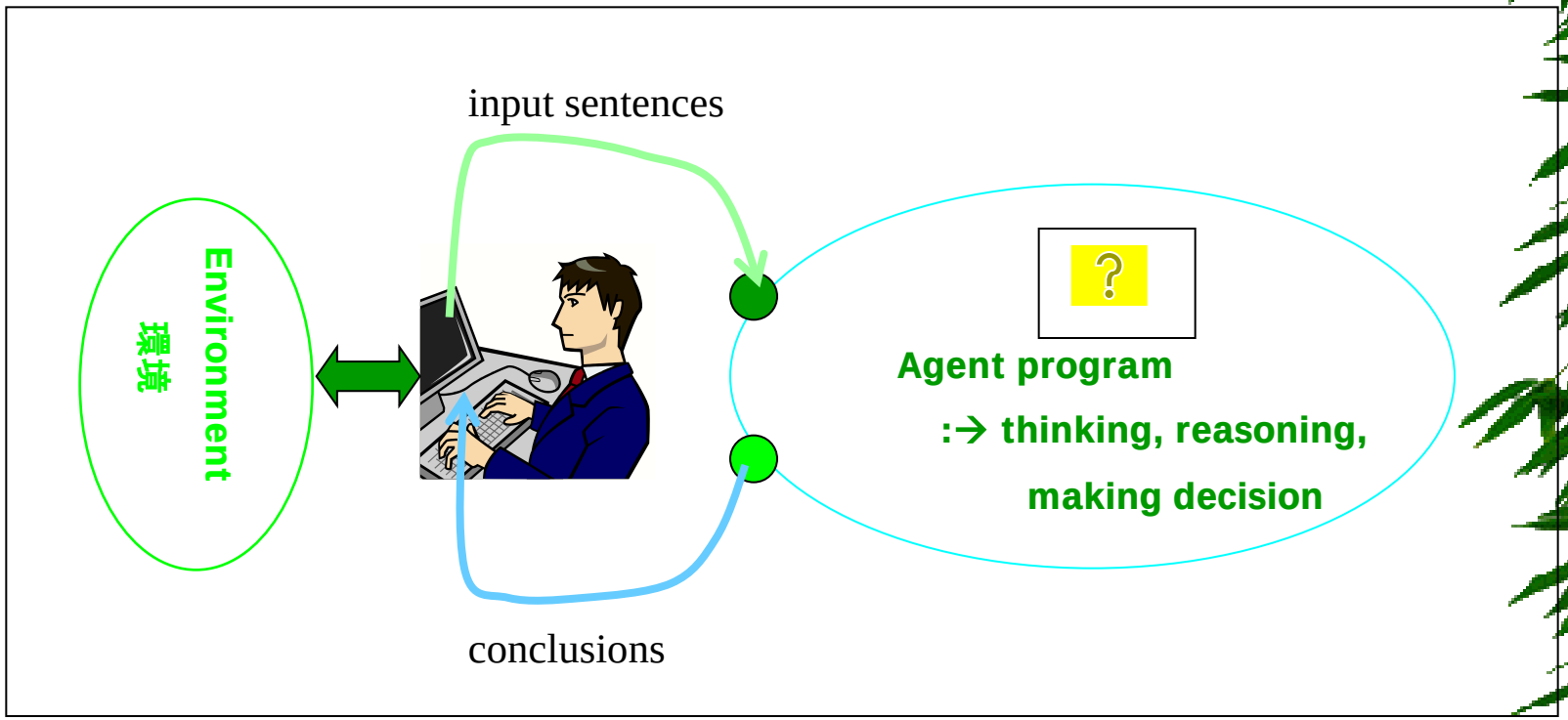
a human driver

a robot driver

a driver based on programs

Fig1. An agent and its environment

For an agent, it takes input sentences and return conclusions



A simple knowledge-based agent

```
function KB-Agent(percept) returns ac action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time
    TELL(KB, Make-Percept-Sentence(percept, t))
    action ←ASK(KB, Make-Action-Query(t))
    TELL(KB, Make-Action-Sentence(action, t))
    t ← t+1
    return action
```

ASK:
Ask a question to KB

TELL:
Add new sentences to KB

Make-Percept-Sentence:

It takes a percept and a time and returns a sentence representing the fact.

Make-Action-Query:

It takes a time as input and returns a sentence.

Make-Action-Sentence:

It takes an action and a time return a sentence representing the action.

How to design an intelligent agent?

- An *agent* perceives its environment via sensors and acts in that environment with its effectors.
- Hence, an agent gets percepts one at a time, and maps this percept sequence to actions (one action at a time).

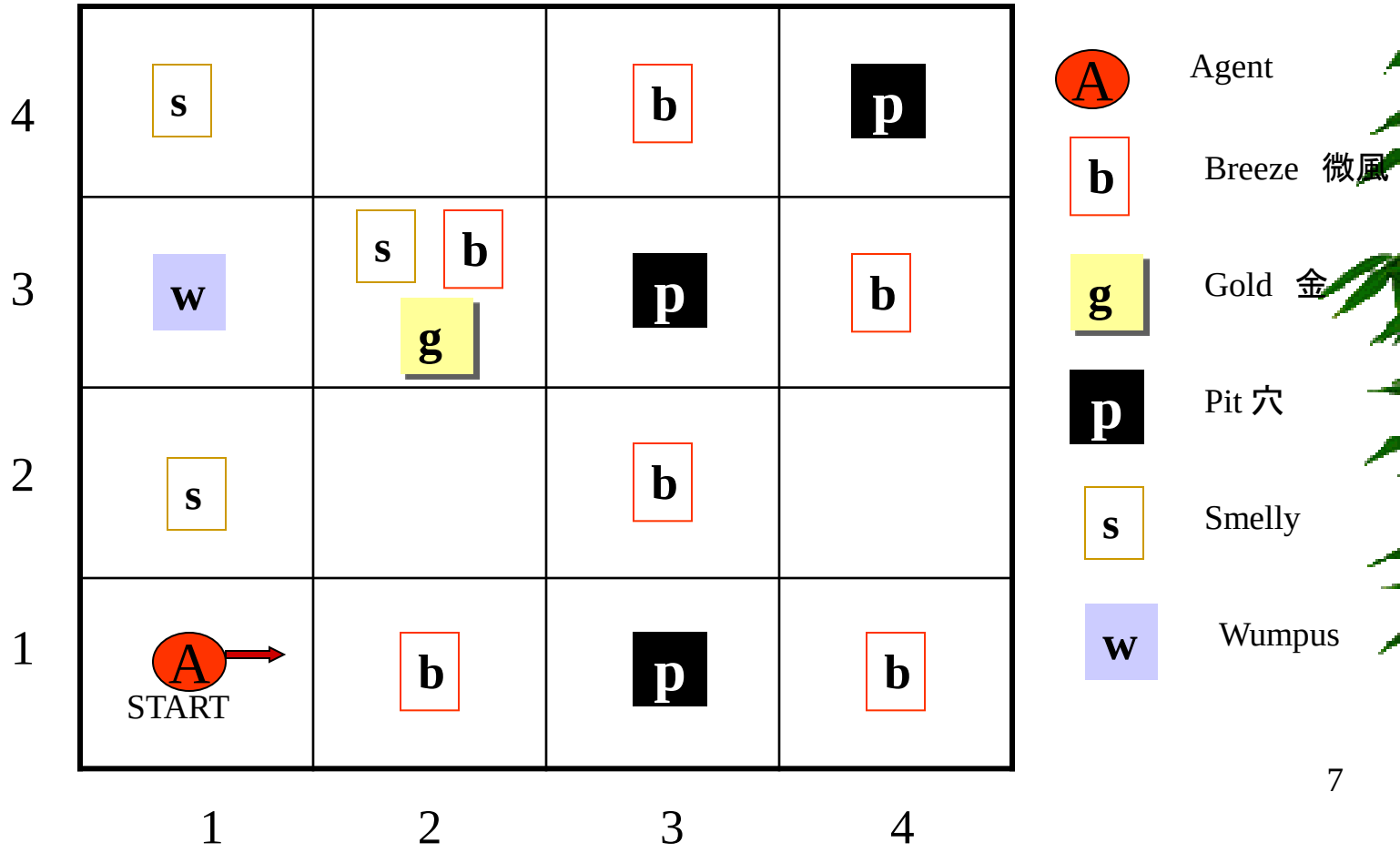
Knowledge about its environment - percepts.

Knowledge of how to operate on percepts.

Q: How to represent "knowledge"?

例： Wumpus world

The wumpus world is a grid of squares surrounded by walls, where each square can contain agents and objects. The agent always starts in the lower left corner, a square that we will label [1,1]. The agent's task is to find the gold, return to [1,1] and climb out of the cave.



PAGE description for designing an intelligent agent

Percepts:	Breeze, Glitter, Smell
Actions:	Left turn, Right turn, Forward, Grab, ….
Goals:	Get gold back to start ….
Environment:	Squares

PAGE: The problem description of Wumpus world

Percepts: Breeze, Glitter, Smell

Actions: Left turn, Right turn, Forward, Grab, Release, Shoot, Climb

Goals: Get gold back to start without entering pit or wumpus square

Environment: squares adjacent to wumpus are smelly

(world) square adjacent to pit are breezy

glitter if and only if gold is in the same square

shooting kills the wumpus if you are facing it

shooting uses up the only arrow

grabbing pick up the gold if in the same square

releasing drops the gold in the same square

climbing leaves the cave

A complete class of environment:

4 x 4 Wumpus world

the agent starts in the square [1, 1] facing towards the right

the location of the gold and the wumpus are chosen randomly with a uniform distribution

(1/15 probability, excluding the start square)

each square can be pit with probability 0.2 (excluding the start sqaure)

Evaluation of agents

Wumpus world characterization

Is the world deterministic? Yes

Is the world fully accessible? No

Is the world static? Yes

Is the world discrete? Yes

Evaluation: 1000 points are awarded for climbing out of cave

100-point penalty for falling into a pit

10000-point penalty for getting killed

What a function is suitable for this problem?

Design in intelligent agent with knowledge -- Knowledge representation

Three levels

The Knowledge level:

One describes the agent by saying what it knows.

The Logic level:

The knowledge is encoded into sentences.

The implementation level:

It is concerned with physical representations of the sentences at the logic level.

Declarative versus learning:

Declarative approach:

to build an agent by telling it what it needs to know

Learning approach:

to design learning mechanisms that output general knowledge about environment given a series of percepts.

How to represent beliefs that can make inferences

initial facts → initial beliefs (knowledge) → inferences → actions

new beliefs ← new facts ←

How to represent a belief (knowledge)? ➔ Knowledge representation

The object of knowledge representation is to express knowledge in computer-tractable form, such that it can be used to help agents perform well.

A knowledge representation language is defined by two aspects:

- Syntax – describes the possible configurations that can constitute sentences.

  defines the sentences in the language

- Semantics – determines the facts in the world to which the sentences refer.

  defines the "meaning " of sentences

Need formal logic language

What is formal logic

- Formal logic is the field of study of entailment relations, formal languages, truth conditions, semantics, and inference.
- All propositions/statements are represented as formulae which have a semantics according to the logic in question.
- **Logical system = Formal language + semantics**
- Formal logics gives us a framework to discuss different kinds of reasoning.

(1) A formal language in which knowledge can be expressed.

$A_{1,1}$

$East_A$

$W_{2,1}$

$\neg$ Forward

(2) Semantics

$A_{1,1} \wedge East_A \wedge W_{2,1}$

(3) A means of carrying out reasoning in such a language.

$A_{1,1} \wedge East_A \wedge W_{2,1} \Rightarrow \neg$ Forward

Sentence:

It is each individual representation.

Knowledge representation language:

It is used for expressing sentences.

Knowledge base:

It is a set of representations of facts about the world.

Logic and inference

Logic

A language is called a **logic** provided the syntax and semantics of the language are defined precisely.

Inference

From the syntax and semantics, an **inference mechanism** for an agent that uses the language can be derived.

- Facts are parts of the world, whereas their representations must be encoded in some way within an agent. All reasoning mechanisms must operate on representations of facts, rather than on the facts themselves.

- The connection between sentences and facts is provided by the semantics of the language. The property of one fact following some other facts is mirrored by the property of one sentence being entailed by some other sentences. Logical inference generates new sentences that are **entailed** by existing sentences.

entail: 〈…を〉必然的に伴う,

Entailment

New sentences generated are necessarily true, given that the old sentences are true. This relation between sentences is called **entailment.**

$$KB \models \alpha$$

Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true.

Here, KB is a set of sentences in a formal language.

For example, $x > 0$ and $y > 0 \models x+y > 0$

Logic roadmap overview

- Propositional logic
- First-order logic
  - Properties, relations, functions, quantifiers, …
  - Terms, sentences, wffs, axioms, theories, proofs, …
- Other logic languages …
- Logical agents
  - Reflex agents
  - Representing change: situation calculus, frame problem
  - Preferences on actions
  - Goal-based agents

Propositional Logic
（命題論理）

Propositional logic: Syntax

- Symbols represent whole propositions (facts).
- The symbols of propositional logic are the logic constants true and false.
- Logic connectives:

  $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implies), and $\Leftrightarrow$ (equivalent)

  If S is a sentence, $\neg S$ is a sentence.

  If S_1 and S_2 is a sentence, $S_1 \wedge S_2$ is a sentence.

  If S_1 and S_2 is a sentence, $S_1 \vee S_2$ is a sentence.

  If S_1 and S_2 is a sentence, $S_1 \Rightarrow S_2$ is a sentence.

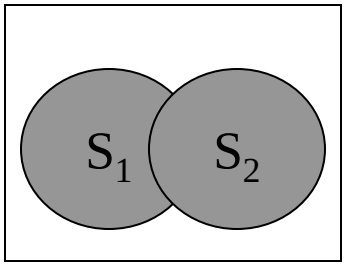
  If S_1 and S_2 is a sentence, $S_1 \Leftrightarrow S_2$ is a sentence.

- The order of the precedence in propositional logic is

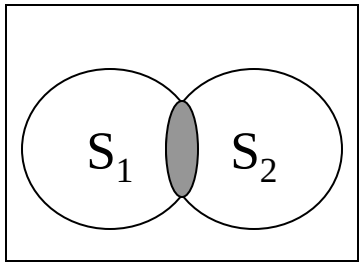
  $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ (from highest to lowest)

Propositional logic: Semantics

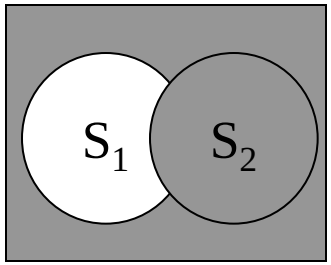
$\neg S$ is true iff	S	is false		
$S_1 \wedge S_2$ is true iff	S_1	is true	**and**	S_2 is true
$S_1 \vee S_2$ is true iff	S_1	is true	**or**	S_2 is true
$S_1 \Rightarrow S_2$ is true iff	S_1	is false	**or**	S_2 is true
i.e., is false iff	S_1	is true	**and**	S_2 is false
$S_1 \Leftrightarrow S_2$ is true iff	$S_1 \Rightarrow S_2$ is true		**and**	$S_2 \Rightarrow S_1$ is true



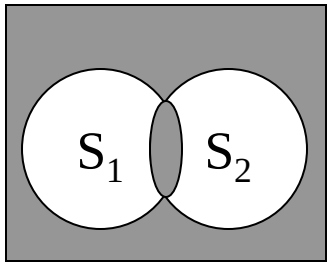
$S_1 \wedge S_2$



$S_1 \vee S_2$



$S_1 \Rightarrow S_2$



$S_1 \Leftrightarrow S_2$

Seven inference rules for propositional Logic

- Modus Ponens $$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

- And-Elimination $$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

- And-Introduction $$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

- Or-Introduction $$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$

- Double-Negation Elimination $$\frac{\neg \neg \alpha}{\alpha}$$

- Unit Resolution $$\frac{\alpha \vee \beta,\ \neg \beta}{\alpha}$$

- Logic connectives: $$\frac{\alpha \vee \beta,\ \neg \beta \vee \gamma}{\alpha \vee \gamma}$$

Inference in Wumpus world（推論）

例:How an agent should act and reason

In the knowledge level

From the **fact** that the agent does not **detect** stench and breeze in [1,1], the agent can **infer** that [1,2] and [2,1] are free of dangers. They are marked OK to indicate this. A cautious agent will only move into a square that it knows is OK. **So** the agent can move forward to [2,1] or turn left 90^0 and move forward to [1,2].

Assuming that the agent first moves forward to [2,1], from the **fact** that the agent **detects** a breeze in [2,1], the agent can **infer** that there must be a pit in a neighboring square, either [2,2][3,1]. **So** the agent turns around (turn left 90^0, turn left 90^0) and moves back to [1,1].

The agent has to move towards to [2,1], from the **fact** that the agent **detects** a stench in [1,2], the agent can **infer** that there must be a wumpus nearby and it can not be in [2,2] (or the agent would have detected a stench when it was in [2,1]). **So** the agent can **infer** that the wumpus is in [1,3]. It is marked with W. The agent can also **infer** that there is no pit in [2,2] (or the agent would detect breeze in [1,2]). **So** the agent can **infer** that the pit must be in [3,1]. After a sequence of deductions, the agent knows [2,2] is unvisited safe square. **So** the agent moves to [2,2].

What is the next move?…… moves to [2, 3] or [3,2]???

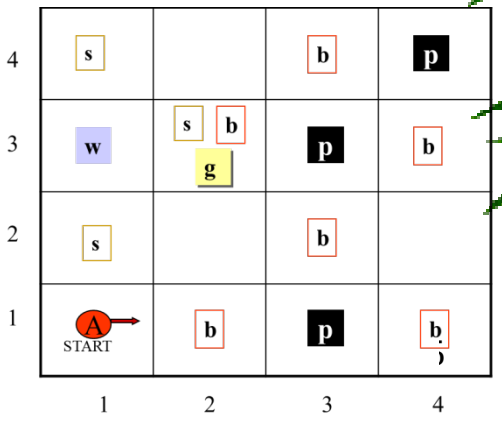
Assuming that the agent moves to [2,3],

from the fact that the agent detects glitter in [2, 3],

agent can infer that there is a gold in [2,3].

So the agent grabs the gold and goes back to the start square

along the squares that are marked with OK.



The knowledge base for Wumpus world problem

Percept sentences:

there is no stench in the square [1,1] ➔ $\neg S_{1,1}$

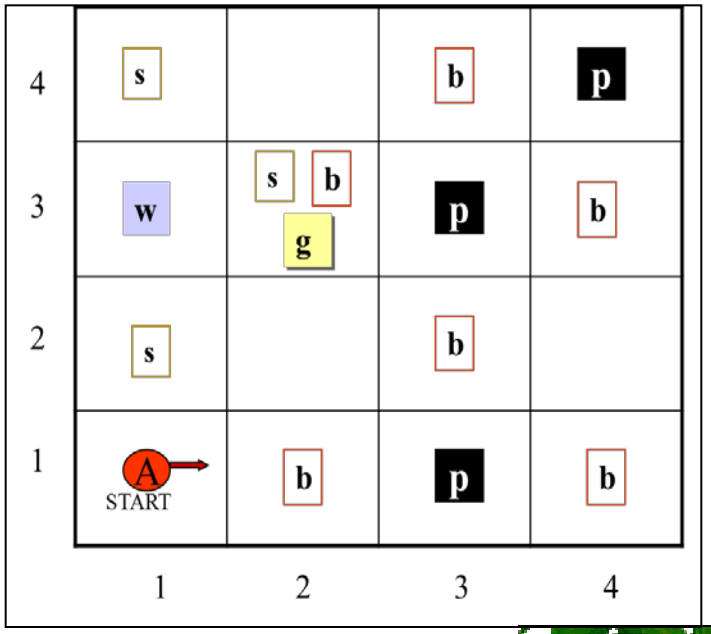
there is no breeze in the square [1,1] ➔ $\neg B_{1,1}$

there is no stench in the square [2,1] ➔ $\neg S_{2,1}$

there is breeze in the square [2,1] ➔ $B_{2,1}$

there is a stench in the square [1,2] ➔ $S_{1,2}$

there is no breeze in the square [1,2] ➔ $\neg B_{1,2}$



knowledge sentences:

if a square has no smell, then neither the square nor any of its adjacent squares can house a wumpus.

R_1: $\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

R_2: $\neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

if there is a stench in [1,2], then there must be a wumpus in [1,2] or in one or more of the neighboring squares.

R_4: $S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$

Inferring knowledge using propositional logic

Concerning with the 6 squares, [1,1], [2,1], [1,2], [3,1], [2,2], [1,3], there are 12 symbols,

$S_{1,1}$, $S_{2,1}$, $S_{1,2}$, $B_{1,1}$, $B_{2,1}$, $B_{1,2}$, $W_{1,1}$, $W_{1,2}$, $W_{2,1}$, $W_{2,2}$, $W_{3,1}$, $W_{1,3}$

The process of finding a wumpus in [1,3] as follows:

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

$$\frac{\alpha \vee \beta, \neg \beta}{\alpha}$$

R_4: $S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$

1. Apply R_1 to $\neg S_{1,1}$, we obtain

   $\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

2. Apply And-Elimination, we obtain

   $\neg W_{1,1}$ $\neg W_{1,2}$ $\neg W_{2,1}$

3. Apply R_2 and And-Elimination to $\neg S_{2,1}$, we obtain

   $\neg W_{1,1}$ $\neg W_{2,2}$ $\neg W_{2,1}$ $\neg W_{3,1}$

4. Apply R_4 and the unit resolution to $S_{1,2}$, we obtain (α is $W_{1,3} \vee W_{1,2} \vee W_{2,2}$ and β is $W_{1,1}$)

   $W_{1,3} \vee W_{1,2} \vee W_{2,2}$

5. Apply the unit resolution again, we obtain (α is $W_{1,3} \vee W_{1,2}$ and β is $W_{2,2}$)

   $W_{1,3} \vee W_{1,2}$

6. Apply the unit resolution again, we obtain (α is $W_{1,3}$ and β is $W_{1,2}$)

   $W_{1,3}$

Here is the answer: the wumpus is in [1,3].

Problem with propositional logic

- Too many propositions ➔

  too many rules to define a competent agent

- The world is changing, propositions are changing with time. ➔

  do not know how many time-dependent propositions we will need

  have to go back and rewrite time-dependent version of each rule.

The problem with proposition logic is that it only has one representational device: the proposition!!!

The solutions to the problem is to introduce other logic

first-order logic

That can represent objects and relations between objects in addition to propositions.

First Logic (述語論理)

First-order Logic

What is first-order logic?

First-order logic makes a stronger set of ontological commitments:

Objects: are things with individual identities and properties.

e.g., people, houses, computers, numbers, Mike Jason, color, …

Properties: are used to distinguish an object from other objects.

e.g., tall, western style, multimedia, prime, English, red, …

Relations: exist and hold among the objects.

e.g., father of, bigger than, made after, equal, student of, …

Functions: are relations in which there is only one "value" for a given "input".

e.g., brother of, increment of, forward, one more than, …

Almost any fact can be thought of as referring to objects and properties or relations/function

. For example:

<u>One plus two equals three.</u>

Objects: one, two, three, one plus one; Relations: equals; Function: plus.

<u>Squares neighboring the wumpus are smelly.</u>

Objects: squares, wumpus; Property: smelly; Relation: neighboring

Syntax of FOL: basic element

- Constant symbols: refer to the same object in the same interpretation

  e.g. Mike Jason, 4, A, B, …

- Predicate symbols: refer to a particular relation in the model.

  e.g., Brother, >,

- Function symbols: refer to particular objects without using their names.

  Some relations are functional, that is, any given object is related to exactly one other object by the relation. (one-one relation)

  e.g., Cosine, FatherOf,

- Variables: substitute the name of an objec.

  e.g., x, y, a, b,… $\forall$x, Cat(x) $\Rightarrow$ Mammal(x)

  if x is a cat then x is a mammal.

- Logic connectives:

  $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implies), and $\Leftrightarrow$ (equivalent)

- Quantifiers: $\forall$ (universal quantification symbol), $\exists$ (existential quantification symbol)

  $\forall$x, for any x, … $\exists$x, there is a x, …

- Equality: = e.g. Father(John) = Henry

Sentences: atomic vs complex

- Atomic sentences = predicate($term_1$, $term_2$, …$term_n$)

  or $term_1$ = $term_2$

  Term = function($term_1$, $term_2$, …$term_n$)

  or constant or variable

  E.g., Sister(Muxin, Yanbo)

  >(height(Muxin), height(Yanbo))

  $\forall$x, >(height(Yanbo), height(x))

- Complex sentences: made from atomic sentences using connectives.

  $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implies), and $\Leftrightarrow$ (equivalent)

  E.g., Sibling(Muxin, Yanbo) $\Rightarrow$ Sibling(Yanbo, Muxin)

  >(Age(Muxin), Age(Yanbo)) $\vee$ $\neg$ >(Age(Muxin), Age(Yanbo))

Truth in first-order logic

- Sentences are true with respect to a **model** and an **interpretation**.
- Model contains objects and relations among them.

E.g., a model: a family.

objects in this model: Robert John, Rose John, Shelly John, Daivd John.

relations: Husband(Robert, Rose), Wife(Rose, Robert),

Father(Robert, David), …, Sister(Shelly, David), …

- Interpretation specifies referents for

constant symbols -> objects

predicate symbols -> relations

function symbols -> functional relations

- An atomic sentence, predicate($term_1$, $term_2$, …$term_n$), is **true**

iff the objects referred to by $term_1$, $term_2$, …$term_n$ are relation referred to by predicate.

E.g, Father(Robert, David) is **true** **iff** Robert and David are two objects in the family model.

Robert is David's father (relation referred to by predicate Father)

Universal quantification

$\forall$<variables> <sentence>

$\forall x$ P is equivalent to the <u>conjunction</u> of <u>instantiations</u> of P

E.g., Every student at CIS is smart:

$\forall x$ At(x, CIS) $\Rightarrow$ Smart(x)

At(Suzuki, CIS) $\Rightarrow$ Smart(x) $\wedge$ At(Abe, CIS) $\Rightarrow$ Smart(x)

$\wedge$ At(Honda, CIS) $\Rightarrow$ Smart(x) $\wedge$ …

Typically, $\Rightarrow$ is the main connective with $\forall$.

Common mistake: using $\wedge$ as the main connective with $\forall$.

$\forall x$ At(x, CIS) $\wedge$ Smart(x)

means Every student is at CIS and every student is smart.

Existential quantification

$\exists$<variables> <sentence>

$\exists$ x P is equivalent to the <u>disjunction</u> of <u>instantiations</u> of P

E.g., Someone at CIS is smart:

$\exists$ x At(x, CIS) $\wedge$ Smart(x)

At(Suzuki, CIS) $\wedge$ Smart(x) $\vee$ At(Abe, CIS) $\wedge$ Smart(x)

$\vee$ At(Honda, CIS) $\wedge$ Smart(x) $\vee$ …

Typically, $\wedge$ is the main connective with $\exists$.

Common mistake: using $\Rightarrow$ as the main connective with $\exists$.

$\exists$ x At(x, CIS) $\Rightarrow$ Smart(x)

is true if there is any student who is not at CIS.

The uniqueness quantifier $\exists!$

E.g., $\exists!$ x King(x) means that there is only one King.

Property of quantifiers

$\forall x \ \forall y$ is the same as $\forall y \ \forall x$

$\exists x \ \exists y$ is the same as $\exists y \ \exists x$

$\exists x \ \forall y$ is not the same as $\exists y \ \forall x$

E.g., $\exists x \ \forall y$ Knows(x, y)

"There is a person who knows everyone in the world"

$\forall y \ \exists x$ Knows(x, y)

"Everyone in the world is known by at least one person"

<u>Quantifier duality</u>: each can be expressed using the other.

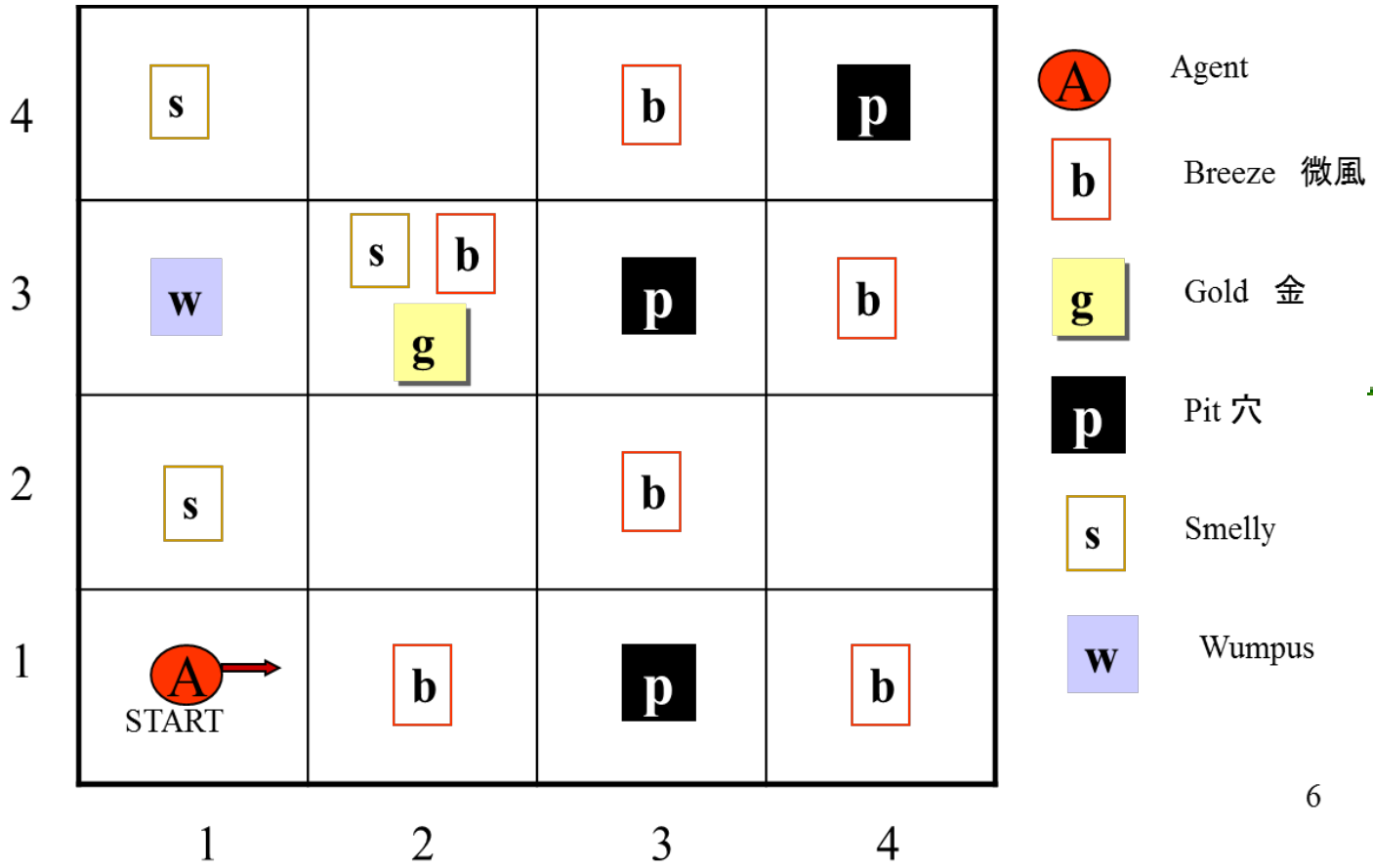
. $\forall x$ Likes(x, iceCream) $\neg \exists x \neg$ Likes(x, iceCream)

"Everyone likes ice cream" means that "There is no one who does not like ice cream"

$\exists x$ Likes(x, carrot) $\neg \forall x \neg$ Likes(x, carrot)

"Someone likes carrot" means that "Not everyone does not like carrot"

Q? Try to describe Wumpus world with first-order logic



Knowledge base for the Wumpus world

<u>Perception:</u> Stench (variable s), Breeze (variable b), Glitter (variable g),

Wall (variable u), Scream (variable v)

$\forall b, g, u, v, t \;\; Percept([S, b, g, u, v], t) \Rightarrow Smelly(t)$

$\forall s, g, u, v, t \;\; Percept([s, B\ g, u, v], t) \Rightarrow Breeze(t)$

$\forall s, b, u, v, t \;\; Percept([s, b, G, u, v], t) \Rightarrow AtGoldRoom(t)$

<u>Reflex:</u> $\forall t \; AtGoldRoom(t) \Rightarrow Action(Grab, t)$

<u>Reflex with internal state:</u> do we have the gold already?

$\forall t \; AtGoldRoom(t) \wedge Holding(\neg Gold, t) \Rightarrow Action(Grab, t)$

Deducing hidden properties

Properties of locations:

$\forall l, t \;\; At(Agent, l, t) \wedge Smell(t) \Rightarrow Smell(l)$

$\forall l, t \;\; At(Agent, l, t) \wedge Breeze(t) \Rightarrow Breeze(l)$

Diagnostic rule – infer cause from effect

e. g. Squares are breezy near a pit

$\forall y \; Breeze(y) \Rightarrow \exists x \;\; Pit(x) \wedge (x=y \vee Adjacent(x, y))$

Causal rule – infer effect from cause

$\forall x, y \; Pit(x) \wedge (x=y \vee Adjacent(x, y)) \Rightarrow Breeze(y)$

……

Keeping track of world change

- Diachronic rules: describe the way in which the world changes.
- Situation calculus: is one way to represent change in FOL.
  - Facts holds in situation rather than eternally.

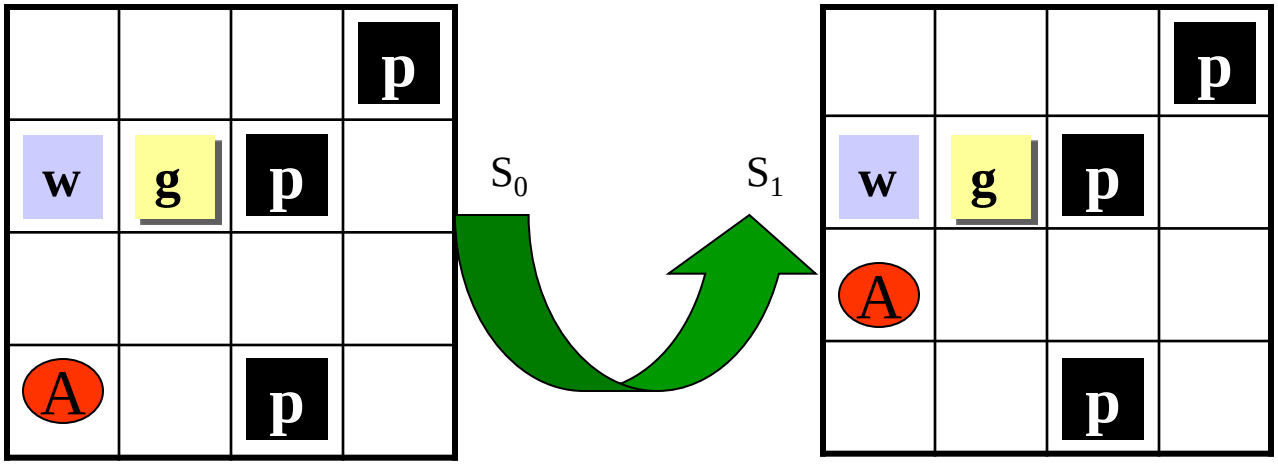
    e.g. Holding(Gold, *Now*) rather than just Holding(Gold)

    Holding(Gold, *Now*) denotes a situation.

    where, *Now* is extra situation argument.
  - Situations are connected by the Result function Result(a, s).

    e.g. At(Agent, [1,1], S_0) $\wedge$ At(Agent, [1,2], S_1)

    Result(Forward, S_0) = S_1



Describing actions

- **Effect axioms** – describe changes due to action

  $\forall$**s** AtGoldRoom(s) $\Rightarrow$ Holding(Gold, Result(Grab, s))

  $\forall$**x, s** $\neg$Holding(x, Result(Release, s))

- **Frame axioms** – describe non-changes due to action

  $\forall$**a, x, s** Holding(x, s) $\wedge$ (a$\neq$Release) $\Rightarrow$ Holding(x, Result(a, s))

- **Successor-state axioms –** combine the effect axioms and the frame axioms

P true afterwards $\Leftrightarrow$ [an action made P true $\vee$ P true already and no action made P false]

$\forall$**a, x, s** Holding(x, Result(a, s)) $\Leftrightarrow$

[((Present(x, s) $\wedge$ a=Grab) $\vee$ (Holding(x, s) $\wedge$ a$\neq$Release)]

Toward a goal

Once the gold is found, the aim now is to return to the start as quickly as possible. Assuming that the agent now is holding “gold ”and has the goal of being at location [1,1].

$\forall$ **s** Holding(Gold, s) $\Rightarrow$ GoalLocation([1,1], s)

The presence of an explicit goal allows the agent to work out a sequence of actions that will achieve the goal. There are at least three ways to find such a sequence:

- **Inference:** to write axioms that will allow us to ASK the KB for a sequence of actions that is guaranteed to achieve the goal safely.
- **Search:** to use a search procedure to find a path to the goal.
- **Planning:** special-purpose reasoning systems designed to reason about actions.

Prolog – a logic language

Prolog プログラム例
(Family relation)

```
parent("Taro", "Ichiro").
parent("Taro", "Jiro").
parent("Taro", "Sakura").
parent("Hanako", "Ichiro").
parent("Hanako", "Jiro").
parent("Hanako", "Sakura").
man("Taro").
man("Jiro").
man("Ichiro").
woman("Hanako").
woman("Sakura").
father(X,Y) :- parent(X,Y), man(X).
mother(X,Y) :- parent(X,Y), woman(X).
daughter(X,Y) :- parent(Y,X), woman(X).
son(X,Y) :- parent(Y,X), man(X).
```

Query
son(X,"Taro").
X = "Ichiro"
X = "Jiro"

Query
mother(X, "Sakura").
X = "Hanako"

Query
father("Taro", "Jiro").
True

Query
parent(X, "Ichiro").

Query
parent("Taro", X).

Others

Logic
Production System
Semantic Net
Frame
Script
Object-Oriented
???

Prolog (Logic programming language) in JavaScript

http://ioctl.org/logic/prolog-latest

Query results

- parent(X, "Ichiro").
- parent("Taro", X).

Enter your query:

parent("Taro", X).

Run query ☐ Show parse results

```
Parsing rulesets.

Attaching builtins to database.
Attachments done.

Parsing query.
X = "Ichiro"

X = "Jiro"

X = "Sakura"
```

Enter your query:

parent(X, "Ichiro").

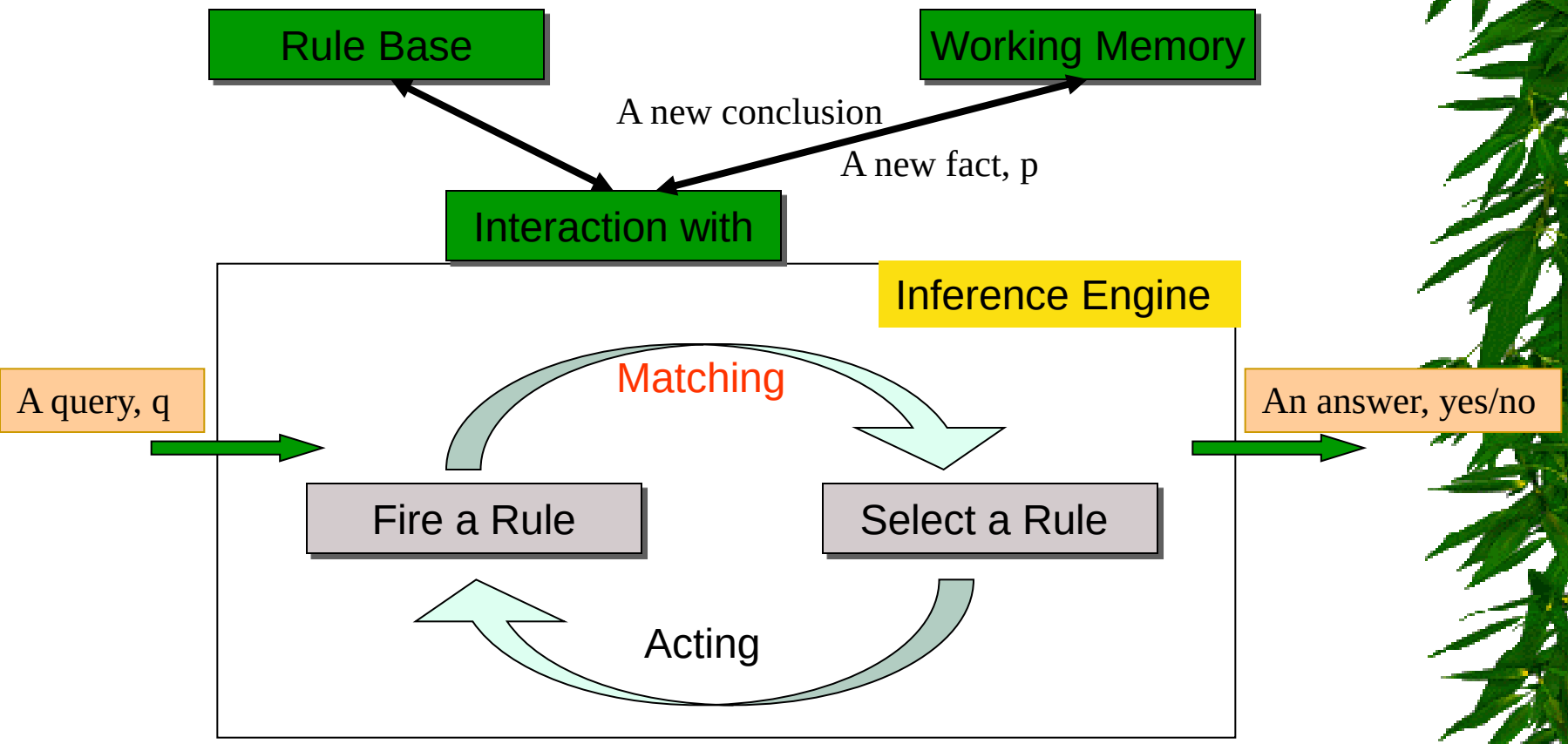
Run query ☐ Show parse results

```
Parsing rulesets.

Attaching builtins to database.
Attachments done.

Parsing query.
X = "Taro"

X = "Hanako"
```

Wumpus world implementation in Prolog

https://archives.limsi.fr/Individu/hernandz/resources/software/wumpus/wumpus.html

A Reasoning System



L6: Reasoning logically - rule-based systems

(Applications of Forward- and Backward-chaining algorithms)

- Review of inference mechanisms
- Rule-based system and its implementation in Java