2.1 Perception and Action

- **Stimulus-response (S-R) agents**
  - Machines that have no internal state and that simply react to immediate stimuli in their environments
  - Based on motor response to rather simple functions of immediate sensory inputs
  - Example: Machina speculatrix, Braitenberg machine
A Robot in a Two-Dimensional Grid World (2)

- Sensory inputs: $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8$
- Robot movements
  - *north* moves the robot one cell up in the cellular grid
  - *east* moves the robot one cell to the right
  - *south* moves the robot one cell down
  - *west* moves the robot one cell to the left
- Division of processes
  - Perception processing and action computation

A Robot in a Two-Dimensional Grid World (3)

- Perceptual processing
  - produces feature vector $X$
    - numeric features: real number
    - categorical features: categories

- Action computation
  - selects an action based on feature vector

A Robot in a Two-Dimensional Grid World (4)

- The split between perception and action is arbitrary
- The split is made in such a way that the same features would be used repeatedly in a variety of tasks to be performed
- The computation of features from sensory signals can be regarded as often used library routines
  - needed by many different action functions
- The next problems
  - (1) converting raw sensory data into a feature vector
  - (2) specifying an action function

Perception

- For the robot task, there are four binary-valued features of the sensory values that are useful for computing an appropriate action $x_1, x_2, x_3, x_4$
- Perceptual processing might occasionally give erroneous, ambiguous, or incomplete information about the robot’s environment
  - Such errors might evoke inappropriate actions
- For robots with more complex sensors and tasks, designing appropriate perceptual processing can be challenging
Action

- Specifying a function that selects the appropriate boundary-following action

  \[
  \begin{align*}
  &\text{if } x_1 = 1 \text{ and } x_2 = 0, \text{ move east} \\
  &\text{if } x_2 = 1 \text{ and } x_3 = 0, \text{ move south} \\
  &\text{if } x_3 = 1 \text{ and } x_4 = 0, \text{ move west} \\
  &\text{if } x_4 = 1 \text{ and } x_1 = 0, \text{ move north}
  \end{align*}
  \]

- None of the features has value 1, the robot can move in any direction until it encounters a boundary.

Boolean Algebra

- Boolean algebra is a convenient notation for representing Boolean functions

  - Rules for Boolean algebra
    - Rules
      - Commutative: \( x \cdot y = y \cdot x, \quad x + y = y + x \)
      - Associative: \( x \cdot (y \cdot z) = (x \cdot y) \cdot z, \quad x + (y + z) = (x + y) + z \)
      - DeMorgan’s law: \( \overline{x \cdot y} = \overline{x} + \overline{y}, \quad \overline{x + y} = \overline{x} \cdot \overline{y} \)
      - Distributive law: \( x \cdot (y + z) = x \cdot y + x \cdot z, \quad x + (y \cdot z) = (x + y) \cdot (x + z) \)

Classes and Forms of Boolean Functions

- A conjunction of literals is a monomial:
  \[ \lambda_1 \lambda_2 \ldots \lambda_k \]
  - The conjunction itself is called a term
  - Bound of the number of monomials of size \( k \) or less:
    \[ \sum_{i=0}^{k} \binom{2n}{i} = \binom{2n+k}{k} \]
  - A clause or a disjunction of literals:
    \[ \lambda_1 + \lambda_2 + \ldots + \lambda_k \]
  - Terms and clauses are duals of each other
  - Disjunctive normal form (DNF): disjunction of terms
    - \( k \)-term DNF: disjunction of \( k \) terms
  - Conjunctive normal form (CNF): conjunction of clauses
    - \( k \)-clause CNF: the size of its largest clause is \( k \)

2.2 Representing and Implementing Action Functions:

Production Systems (1)

- Production system comprises an ordered list of rules called production rules or productions

  - \( c \rightarrow a \), where \( c \) is the condition part and \( a \) is the action part
  - Production system consists of a list of such rules
  - Condition part
    - Can be any binary-valued function of the features
    - Often a monomial
  - Action part
    - Primitive action, a call to another productive system, or a set of actions to be executed simultaneously
Production Systems (2)

- Production system representation for the boundary following routine
  - An example of a durative systems-system that ends

- Teleo-reactive (T-R) programs
  - Each properly executed action in the ordering works toward achieving a condition higher in the list
  - Usually easy to write, given an overall goal for an agent
  - Quite robust: actions proceed inexorably toward the goal
  - Can have parameters that are bound when the programs are called
  - Can call other T-R programs and themselves recursively

Networks (1)

- Threshold logic unit (TLU)
  - Circuit consists of networks of threshold elements or other elements that compute a nonlinear function of a weighted sum of their inputs

- Linearly separable functions
  - The boolean functions implementable by a TLU
  - Many boolean functions are linearly separable
  - Exclusive-or function of two variables is an example of not linearly separable

Networks (2)

- An implementation of the boundary following production rule
  - $x_1 \cdot x_2 \rightarrow \text{north}$
  - $x_2 \cdot x_4 \rightarrow \text{west}$
  - $x_2 \cdot x_3 \rightarrow \text{south}$
  - $x_1 \cdot x_2 \rightarrow \text{east}$
  - $1 \rightarrow \text{north}$

- Neural network
  - Network of TLUs
  - For more complex problems
  - TLUs are thought to be simple models of biological neurons
  - Connection weights
  - Threshold value

Networks (3)

- A simple network structure with repeated combination of inverters and AND gates can be used to implement any T-R program
Networks (4)

- **TISA (Test, Inhibit, Squelch, Act)**
  - Each rule in the T-R program is implemented by a subcircuit (called a TISA) with two inputs and two outputs.
  - One TLU in the TISA computes the conjunction of one of its input with the complement of the other input; the other TLU computes the disjunction of its two inputs.
  - The inhibit input is 1 when none of the rules above has a true condition.
  - The test input is 1 only if the condition $C_i$ corresponding to this rule is satisfied.
  - The act output is 1 when the test input is 1 and the inhibit input is 0.
  - The squelch output is 1 when either the test input or the inhibit input is 1.

The Subsumption Architecture (1)

- Proposed by Rodney Brooks
- The general idea: An agent’s behavior is controlled by a number of “behavior modules.”

The Subsumption Architecture (2)

- If the sensory inputs satisfy a precondition specific to that module, then a certain behavior program, also specific to that module, is executed.
- One behavior module can subsume another.
- Complex behaviors can emerge from the interaction of a relatively simple reactive machine with complex environment.