Appendix A
Logico-Mathematical Background

J. C. Huang
Department of Computer Science
University of Houston

This chapter contains materials useful for

- interpreting the program specification correctly,
- determining if any part of the program specification is violated (i.e., not satisfied),
- proving that a certain assertion is a theorem (i.e., always true),
- arguing for (or against) the correctness of a given program,
- relating a subfunction in the specification to the corresponding subprogram, and
- finding an input to test-execute a specific subprogram.

Motivating question:

Suppose the specification of the payroll program that is used to print your paycheck contains the following statement:

"*If the employee is a US citizen, compute the social security tax and deduct it from the gross amount.*"

Now suppose that you find an instance of social-security-tax deduction on the pay check of a non-citizen. Can you conclude that the program is incorrect?

Another motivating question:

Suppose the requirements of a computerized decision support system include the following statement:

> *"If the drug passes both animal test and clinical test, then the company will market it if and only if it can be produced and sold profitably and the government does not intervene."*

Now further suppose that the product failed to pass the clinical test and this system supports the company to market the product. Does the system violate this requirement?

The Propositional Calculus

A *proposition* is a declarative sentence that is either true or false. For example,

- Harvard is a private university.
- $x + y = y + x$
- Eleven is divisible by three.
- The number 4 is a prime number.

are propositions. The first two sentences are true, and the last two false.

Connectives

Given propositions, we can form new propositions by combining them with connectives such as

- negation: $\neg$, not
- conjunction: $\wedge$, and
- disjunction: $\vee$, or
- implication: $\supset$, implies, if ... then ...
- equivalence: $\equiv$, ... if and only if ...

Definition of the connectives:

p	q	¬p	p ∧ q	p ∨ q	p ⊃ q	p ≡ q
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Well-formed formula

A *well-formed formula* (wff) in the language of the propositional calculus is a syntactically correct expression. It is composed of connectives, propositional variables (such as p, q, r, s, ...), constants (T and F), and parentheses.

The syntax

The syntax of a wff is recursively defined as follows

1. A propositional variable standing alone is a wff.
2. If is a wff then ¬(α) is also a wff.
3. If and are wffs then (α) ∧ (β), (α) ∨ (β), (α) ⊃ (β), and (α) ≡ (β) are also wffs.
4. Those and only those obtainable by a finite number of applications of (1), (2), and (3) are wffs.

The precedence relation:

A wff obtained by the above definition may contain many parentheses and thus not suitable for human consumption. The use of parentheses can be reduced by using the following precedence (listed in the descending order):

$\neg,\ \wedge,\ \vee,\ \supset,\ \equiv$

Truth Table

The *truth table* of a wff lists the truth values of the formula for all possible combinations of assignments to the values of variables involved. For example:

p	q	$p \supset q$	$p \wedge (p \supset q)$
F	F	T	F
F	T	T	F
T	F	F	F
T	T	T	T

Practical application

- In practice, a rule or specification is stated as a proposition.
- A rule is not violated, or a specification is satisfied, if it is evaluated to be true.
- Analysis of a statement can often be facilitated by translating it into a well-formed formula first.

Example A.1

Suppose the policy of a pharmaceutical company includes the following statement:

Proposition Alpha: *If the drug passes both animal test and clinical test, then the company will market it if and only if it can be produced and sold profitably and the government does not intervene.*

Example A.1 (continued)

Now let us further suppose that the company is developing a new drug with an enormous market potential, and an ambitious manager has just decided to put the drug on the market immediately, despite of the fact that the drug has failed the clinical test. Does this decision to market the drug violate the policy stated as Proposition Alpha?

Example A.1 (continued)

First, translate the above statement into the well-formed formula shown below:

$$A_1: \quad a \wedge c \supset (m \equiv p \wedge \neg g)$$

where

a: the drug passes animal test
c: the drug passes clinical test
m: the company will market the drug
p: the drug can be produced and sold profitably
g: the government intervenes

Example A.1 (continued)

It is obvious that, if the drug failed the clinical test, i.e., if c is false, then the formula is true regardless of the truth values of other variables.

$$a \wedge c \supset (m \equiv p \wedge \neg g)$$

$$? \; F \; F \; T \qquad\qquad ?$$

Hence the policy is not violated.

Example A.1 (continued)

Note, however, that the policy is ambiguous in that it can also be interpreted as follows:

$$A_2: \quad a \wedge c \supset m \equiv p \wedge \neg g.$$

$$F \; F \; T \quad ? \; ? \quad ?$$

If the policy is interpreted this way then there is insufficient information to determine if the policy is violated.

Tautology and contradiction

Definition A.2: If for every assignment of values to its variables a wff has the value T, it is said to be a *tautology;* if it always has the value F then it is said to be *contradictory* (or, a *contradiction*). A wff is said to be *satisfiable* if and only if it is not contradictory. A wff is said to be *contingent* if and only if it is neither valid nor contradictory.

Tautology and contradiction

Notation A.3: If A is a tautology, we write |-A.

Note that A is a tautology if and only if ¬A is a contradiction.

Relations among propositions

In practice, we often have to deal with a set of propositions, and it is useful to define the following two relations among them.

Logical equivalence

Definition A.4: Two wffs A and B are said to be *logically equivalent* if and only if they have the same truth table.

Theorem A.5: A and B are logically equivalent if and only if |-A ≡ B.

Logical consequence

Definition A.6: B is a *logical consequence* of A (denoted by A |- B) if for each assignment of truth values to the variables of A and B such that A has the value T then B also has the value T.

Theorem A.7: A |- B if and only if |-A ⊃ B.

Logical consequence

The antecedent A may consist of a set of propositions. In that case, we have

Theorem A.8: $A_1, A_2, \ldots, A_n \vdash B$ if and only if

$$\vdash A_1 \wedge A_2 \wedge \ldots \wedge A_n \supset B.$$

How to prove that A ⊃ B is a theorem?

By the definition of the implication (⊃) connective, A ⊃ B can be false only if A is true and B is false.

Hence, a common technique for showing |-A ⊃ B is to show that A cannot be true if B is false.

Multiple Antecedents

In practical applications, the antecedent often consists of a conjunction of n parts: A_1, A_2, … , A_n. In that case,

- to show that the assumption is consistent is to show that$A_1 \wedge A_2 \wedge \ldots \wedge A_n$ is satisfiable, and
- to show that $A_1 \wedge A_2 \wedge \ldots \wedge A_n \supset B$ is a tautology is to let B be false and show that it is impossible to make all A_i's true at the same time.

Applications

In applications where we have to make a decision to satisfy a number of constraints, a relation of the type $\vdash A_1 \wedge A_2 \wedge \ldots \wedge A_n \supset B$ is useful. Let A_i's be the constraints and B be the proposed decision. B is a valid decision if

$\vdash A_1 \wedge A_2 \wedge \ldots \wedge A_n \supset B$ because

- Decision B leads to satisfaction of all constraints.
- Decision to the contrary (i.e., $\neg B$) leads to violation of a rule or contradiction of a fact.

Applications (continued)

On the other hand, if B is not a logical consequence of A_i's, a decision to the contrary, i.e., $\neg B$, could be a valid decision because in that case it is possible to make B false without violating any A_i.

Example

Now let us get back to Proposition Alpha that says:

"If the drug passes both animal test and clinical test, then the company will market it if and only if it can be produced and sold profitably and the government does not intervene"

which can be translated into a wff as

$$A_2: \quad a \wedge c \supset m \equiv p \wedge \neg g$$

Example

Let us suppose that, in addition to the policy expressed as Proposition Alpha, which is repeated below for convenience.

A_1 or A_2:

If the drug passes both animal test and clinical test, then the company will market it if and only if it can be produced and sold profitably and the government does not intervene.

Example (continued)

the company further stipulates that *if the drug cannot be produced and sold profitably, the company should not market it.*

This can be expressed as a wff as

$$A_3: \quad \neg p \supset \neg m$$

Example (continued)

Furthermore, the company requires its decision makers to keep in mind that *if the drug failed the animal test or clinical test, and the drug is marketed, the government will intervene.*

This can be expressed as

$$A_4: \quad (\neg a \vee \neg c) \wedge m \supset g$$

Example (continued)

Also remember that the drug failed the clinical test, i.e.,

A_5: $\neg c$

Now if we can show that $\vdash A_2 \wedge A_3 \wedge A_4 \wedge A_5 \supset B$, where B is $\neg m$, it means that all constraints can only be satisfied at the same time by not marketing the drug.

That is to say, the company cannot market the drug without violating at least one rule or contradicting a fact.

Example (continued)

In other words, if we let

A_2: $a \wedge c \supset m \equiv p \wedge \neg g$

A_3: $\neg p \supset \neg m$

A_4: $(\neg a \vee \neg c) \wedge m \supset g$

A_5: $\neg c$

B: $\neg m$

Is $A_2 \wedge A_3 \wedge A_4 \wedge A_5 \supset B$ a tautology?

Example (continued)

It turns out that $A_2 \wedge A_3 \wedge A_4 \wedge A_5 \supset B$ is a tautology.

It can be proved by letting B to be false, and see if we can make all A_i's true at the same time as demonstrated in the following.

Example (continued)

1. Let m←T so that B becomes false.

B: ¬ m

Example (continued)

1. Let m←T so that B becomes false.
2. Let c←F to make A_5 true.

A_5: ¬ c

Example (continued)

1. Let m⟵T so that B becomes false.
2. Let c⟵F to make A_5 true.
3. (1) and (2) requires that g⟵T to make A_4 true.

A_4: $(\neg a \vee \neg c) \wedge m \supset g$

Example (continued)

1. Let m←T so that B becomes false.
2. Let c←F to make A_5 true.
3. (1) and (2) requires that g←T to make A_4 true.
4. This requires that p←T to make A_3 true.

A_3: $\neg p \supset \neg m$

Example (continued)

1. Let m←T so that B becomes false.
2. Let c←F to make A_5 true.
3. (1) and (2) requires that g←T to make A_4 true.
4. This requires that p←T to make A_3 true.
5. For A_2 to be true, we need to set g←F. This contradicts (3).

$$A_2: \quad a \wedge c \supset m \equiv p \wedge \neg g$$

Example (continued)

1. Let m←T so that B becomes false.
2. Let c←F to make A_5 true.
3. (1) and (2) requires that g←T to make A_4 true.
4. This requires that p←T to make A_3 true.
5. For A_2 to be true, we need to set g←F. This contradicts (3).
6. Hence $A_2 \wedge A_3 \wedge A_4 \wedge A_5 \supset B$ is a tautology.

Example (continued)

That is to say, in views of the three rules/policies and the fact that the drug has failed the clinical test, the company should not market the drug unless the company is willing to violate at least one of the rules or contradict the fact that the drug failed the clinical test.

Example (continued)

Note that A_2, A_3, A_4, and A_5 are consistent in that it is possible to make them all true at the same time, e.g. by using the following assignment:

$c \leftarrow F$

$m \leftarrow F$

$p \leftarrow T$

$g \leftarrow F$

$a \leftarrow T$

Example (continued)

Also note that $A_1 \wedge A_3 \wedge A_4 \wedge A_5 \supset B$ is not a tautology (prove this!). That is to say, if the first rule is interpreted in the way denoted by A_1, the company may decide to market or not to market the drug without violating any rule or contradicting any fact denoted by A_1, A_3, A_4, and A_5.

The First-Order Predicate Calculus

An extension of the propositional calculus that allows us to deal with sentences of the form:

$x > 0$

She is six feet tall.

Everyone is dressed in blue today.

These are called *sentential forms*.

Additional vocabulary

In addition to those in the propositional calculus, the first-order predicate calculus includes symbols

for individual constants (names of individuals): a, b, c, …

for individual variables (pronouns): x, y, z, …

for function letters (to denote functions): f, g, h, …

for predicate letters (to denote predicates): F, G, H, …

for quantifiers: universal quantifier $(\forall x)$,

existential quantifier $(\exists x)$.

The syntax

Definition A.9: A *term* is defined as follows:

(1) Individual constants and individual variables are terms.

(2) If f is an n-ary functional letter and $t_1, t_2, \ldots, t_n$ are terms then $f(t_1, t_2, \ldots, t_n)$ is a term.

(3) Those and only those obtained by (1) and (2) are terms.

The syntax (continued)

Definition A.10: A string is an *atomic formula* if it is

(1) a propositional variable standing alone, or

(2) a string of the form $F(t_1, t_2, \ldots, t_n)$, where F is an n-ary predicate letter and $t_1, t_2, \ldots, t_n$ are terms.

The syntax (continued)

Definition A.11: a *well-formed formula (wff)* in the language of the first-order predicate calculus is defined as follows.

(1) An atomic formula is a wff.

(2) If A is a wff and x is an individual variable then $(\forall x)A$ and $(\exists x)A$ are wff.

(3) If A and B are wffs the $\neg A$, $(A) \wedge (B)$, $(A) \vee (B)$, $(A) \supset (B)$, and $(A) \equiv (B)$ are wffs.

(4) Those and only those obtained by 1, 2, and 3 are wffs.

The quantifiers

The notation

$(\forall x)P$ is to be read as "for all x (in the domain) … " and

$(\exists x)P$ is to be read as "there exists an x (in the domain) such that …".

The scope of a quantifier

is the subexpression to which the quantifier is applied. The occurrence of an individual variable, say, x, is said to be *bound* if it is either an occurrence $(\forall x)$, $(\exists x)$, or within the scope of a quantifier $(\forall x)$ or $(\exists x)$. Any other occurrence of a variable is a *free* occurrence.

Example:

$$P(x) \wedge (\exists x)(Q(x) \equiv (\forall y)R(y))$$

More about *scope*

A variable may be within the scope of more than one quantifier.

In that case, an occurrence of a variable is bound by the innermost quantifier on that variable within whose scope that particular occurrence lies.

Interpretation

Definition A.12: An *interpretation* of a wff consists of a non-empty domain D and an assignment to each n-ary predicate letter of an n-ary predicate on D, to each n-ary function letter of an n-ary function on D, and to each individual constant of a fixed element of D.

Satisfiability in a domain

Definition A.13: A wff is *satisfiable* in a domain D if there exists an interpretation with domain D and assignments of elements of D to the free occurrences of individual variables in the formula such that the resulting proposition is true.

Validity in a domain

Definition A.14: A wff is *valid* in a domain D if for every interpretation with domain D and assignment of elements of D to free occurrences of individual variables in the formula the resulting proposition is true.

Satisfiability and Validity

A wff is *satisfiable* if it is satisfiable in some domain.

A wff is *valid* if it is valid in all domains.

Example A.15

Consider the wff $(\forall x)P(f(x, a), b)$. A possible interpretation of this wff would be

D: the set of all integers

P(u, v): $u > v$

f(y, z): $y + z$

a: 1

b: 0

That is, it becomes $(\forall x)_D(x + 1 > 0)$, and is false.

Example A.16

Example: Consider the wff $(\forall x)(\exists y)P(f(x, y), a)$. A possible interpretation of this wff would be:

D: the set of all integers

P(u, v): u is equal to v

f(x, y): x + y

a: 0

Thus it reads $(\forall x)(\exists y)(x + y = 0)$, and is true.

The order in which quantifiers occur

Observe that the order in which the quantifiers are given is important, and cannot be arbitrarily changed. For example,

$(\forall x)(\exists y)(x + y = 0)$ is true, and
$(\exists x)(\forall y)(x + y = 0)$ is false

in the previous interpretation.

Prenex normal form

Definition A.18: A wff is said to be in the *prenex normal form* if it is of the form

$$(Q_1x_1)(Q_2x_2) \ldots (Q_nx_n)M,$$

where each (Q_ix_i) is either $(\forall x_i)$ or $(\exists x_i)$, and M is a formula containing no quantifiers.

$(Q_1x_1)(Q_2x_2) \ldots (Q_nx_n)$ is called the *prefix* and M the *matrix* of the formula.

Theorems A.17

$(\exists x)(\exists y)A \equiv (\exists y)(\exists x)A$

$(\forall x)(\forall y)A \equiv (\forall y)(\forall x)A$

$(\forall x)(A \supset B) \equiv ((\exists x)A \supset B)$

$(\exists x)(A \supset B) \equiv ((\forall x)A \supset B)$

$(\forall x)(A \supset B) \equiv (A \supset (\forall x)B)$

$(\exists x)(A \supset B) \equiv (A \supset (\exists x)B)$

where x does not occur free in B in (3) and (4), and in A in (5) and (6).

Theorems (continued)

To illustrate the necessity of the qualifier "where x does not occur free in B" for (3), let us consider the following interpretation where x occurs free in B(x, y):

D: the set of all positive integers

A(x, y): x divides y

B(x, y): $x \leq y$

Theorems (continued)

With this interpretation (3) reads

$$(\forall x)(\text{"x divides y"} \supset x \leq y)$$

$$\equiv (\exists x)(x \text{ divides } y) \supset x \leq y.$$

Although the left-hand side of the "≡" is true, the truth value of the right-hand side depends on the assignment made to the free variable x, and thus the equivalence relation does not hold.

Theorems (continued)

Now if we interpret B(x, y) to be $(\exists x)((y \div x)x=y)$, (3) reads

$$(\forall x)(\text{"x divides y"} \supset (\exists x)((y \div x)x=y))$$

$$\equiv (\exists x)(x \text{ divides } y) \supset (\exists x)((y \div x)x=y).$$

The equivalence relation holds because x does not occur free in B.

Theorems (continued)

Note that the equivalence relation also holds if x does not occur in B at all. For example, if we interpret B(x, y) to be "y is not prime" then (3) reads

$$(\forall x)(\text{"x divides y"} \supset \text{"y is not prime"})$$
$$\equiv (\exists x)(x \text{ divides } y) \supset \text{"y is not prime".}$$

More theorems (for ∧ and ∨)

(1a) $\neg((\exists x)A(x)) \equiv (\forall x)(\neg A(x))$

(1b) $\neg((\forall x)A(x)) \equiv (\exists x)(\neg A(x))$

(2a) $(Qx)A(x) \vee B \equiv (Qx)(A(x) \vee B)$

(2b) $(Qx)A(x) \wedge B \equiv (Qx)(A(x) \wedge B)$

(3a) $(\exists x)A(x) \vee (\exists x)C(x) \equiv (\exists x)(A(x) \vee C(x))$

(3b) $(\forall x)A(x) \wedge (\forall x)C(x) \equiv (\forall x)(A(x) \wedge C(x))$

(4a) $(Q_1x)A(x) \vee (Q_2x)C(x) \equiv (Q_1x)(Q_2y)(A(x) \vee C(y))$

(4b) $(Q_3x)A(x) \wedge (Q_4x)C(x) \equiv (Q_3x)(Q_4y)(A(x) \wedge C(y))$

where x does not occur in B in (2), y does not occur in A(x) in (4), and Q, Q_1, Q_2, Q_3, and Q_4 are either $\exists$ or $\forall$.

Example A.19

$(\forall x)P(x) \wedge (\exists x)Q(x) \vee \neg(\exists x)R(x)$

$(\forall x)P(x) \wedge (\exists x)Q(x) \vee (\forall x)(\neg R(x))$ by (1a)

$(\forall x)(\exists y)(P(x) \wedge Q(y)) \vee (\forall x)(\neg R(x))$ by (4b)

$(\forall x)(\exists y)(\forall z)(P(x) \wedge Q(y) \vee \neg R(z))$ by (4a)

Example application

Find an assignment that satisfies

$$b - a > e \ \wedge \ b + 2a \geq 6 \ \wedge \ 2(b - a)/3 \leq e$$

which is the path condition of an execution path.

Inequalities expressed as equalities

$a \neq b$ if and only if there exists an $x \neq 0$ such that $a + x = b$.

$a < b$ if and only if there exists an $x > 0$ such that $a + x = b$.

$a \leq b$ if and only if there exists an $x \geq 0$ such that $a + x = b$.

$a > b$ if and only if there exists an $x < 0$ such that $a + x = b$.

$a \geq b$ if and only if there exists an $x \leq 0$ such that $a + x = b$.

Alternative expressions

In the language of the predicate calculus:

$$a = b \Leftrightarrow (\exists x)_{\neq 0}(x = b - a)$$
$$a < b \Leftrightarrow (\exists x)_{> 0}(x = b - a)$$
$$a \leq b \Leftrightarrow (\exists x)_{\geq 0}(x = b - a)$$
$$a > b \Leftrightarrow (\exists x)_{> 0}(x = a - b)$$
$$a \geq b \Leftrightarrow (\exists x)_{\geq 0}(x = a - b)$$

The path condition

can be expressed as

$$(\exists x)_{D_1}(x = b - a - e)$$

$$\wedge\ (\exists x)_{D_2}(x = b + 2a - 6)$$

$$\wedge\ (\exists x)_{D_2}(x = e - 2(b - a)/3)$$

where D_1 is the set of all real numbers >0

and D_2 is the set of all real numbers ≥ 0

The path condition

in its prenex normal for is found to be

$$(\exists x)_{D_1}(\exists y)_{D_2}(\exists z)_{D_2}(x = b - a - e \wedge y = b + 2a - 6 \wedge z = e - 2(b - a)/3)$$

which can be reduced to:

$$(\exists x)_{D_1}(\exists y)_{D_2}(\exists z)_{D_2}(3x - y + 3z = 6 - 3a)$$

Test-case selection

The domain of this formula requires that the value of x, y, and z must satisfy $x > 0$, $y \geq 0$, and $z \geq 0$. Now if we let

$x \leftarrow 0.1, \quad y \leftarrow 0, \quad z \leftarrow 0.$

then a possible assignment would be

$a \leftarrow 1.9 \quad b \leftarrow 2.2 \quad e \leftarrow 0.2$

that satisfies the path condition

$b - a > e \ \wedge \ b + 2a \geq 6 \ \wedge \ 2(b - a)/3 \leq e$

Principle of Mathematical Induction

If 0 has a property P, and if any integer n is P then n+1 is also P, then every integer is P.

The principle is used in proving statements about integers or, derivatively, in proving statements about sets of objects of any kind which can be correlated with integers.

Steps involved in constructing a proof

The procedure is to prove that

(a) 0 is P (induction basis),

to assume that

(b) n is P (induction hypothesis),

to prove that

(c) n+1 is P (induction step)

using (a) and (b); and then to conclude that

(d) n is P for all n.

Example

Suppose we wish to prove that

$$\sum_{i=0}^{n} i = 0 + 1 + 2 + \ldots + n = n(n+1)/2$$

Example (continued)

To begin, we must state the property that we want to prove. This statement is called the induction proposition. In this case P is directly given by

$$n \text{ is } P \Leftrightarrow \sum_{i=0}^{n} i = n(n+1)/2$$

Example (continued)

(a) For the basis of the induction we have, for n = 0,
0 = 0(0+1)/2, which is true.

(b) The induction hypothesis is that k is P for some arbitrarily choice of k:

$$\sum_{i=0}^{k} i = 0 + 1 + 2 + \ldots + k = k(k+1)/2$$

Example (continued)

(c) For the induction step, proving k + 1 is P, we have

$$\sum_{i=0}^{k+1} i = \sum_{i=0}^{k} i + (k+1) = (k+1)/2 + (k+1)$$

$$= k^2 + 3k + 2/2$$

$$= (k+1)(k+2)/2$$

$$= (k+1)((k+1)+1)/2$$

(using the induction hypothesis)

Example (continued)

(d) Hence k+1 has the property P.

Inductive (or recursive) definition

Inductive definition of a set or property P:

given a finite set A,

(a) the elements of A are P (basis clause)

(b) the elements of B, all of which are constructed from A, are P (inductive clause)

(c) the elements constructed as in (a) and (b) are the only elements of P (extremal clause)

Recursive definition of a set

say, D

1. Element d_0 is in D.
2. If d is in D and P(d) then f(d) is also in D.
3. Those and only those obtained by a finite applications of (1) and (2) are in D.

Example

Recursive definition of set D of even integers between 0 and 32, inclusive:

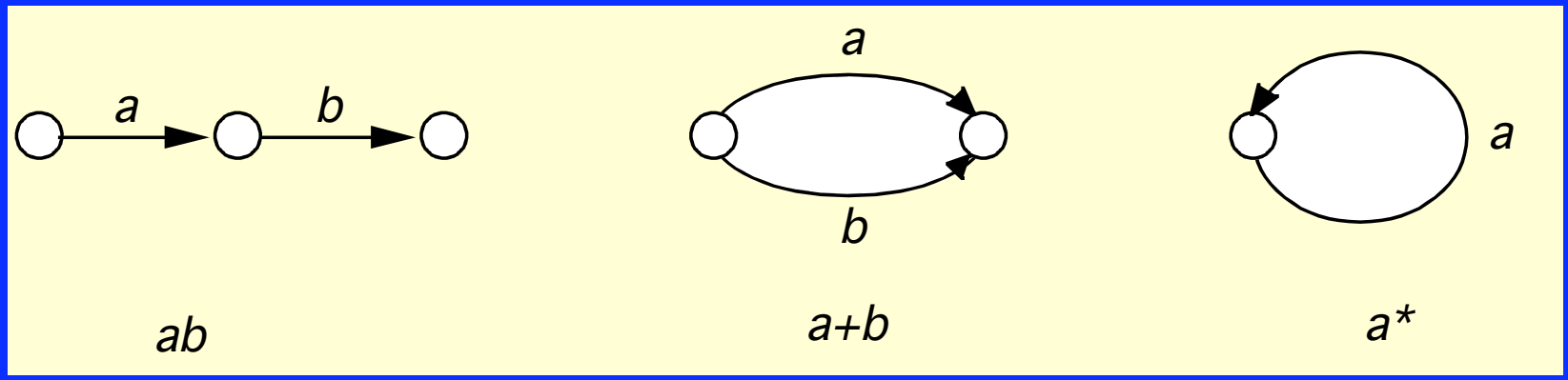
1. 0 is an element of D.
2. If $d \in D$ and $d \leq 30$ then $d+2$ is also an element in D.
3. Those and only those obtained by a finite application of (1) and (2) are elements of D.

Directed Graphs and Path Description

When we study the logical structure of a program, we need to be able to speak of a path structure precisely and concisely. This can be accomplished by making use of the language of regular expressions.

Three basic connection schemes

A set of paths between any two nodes in a (directed) graph can be described in terms of symbols associated with the constituent edges as follows.

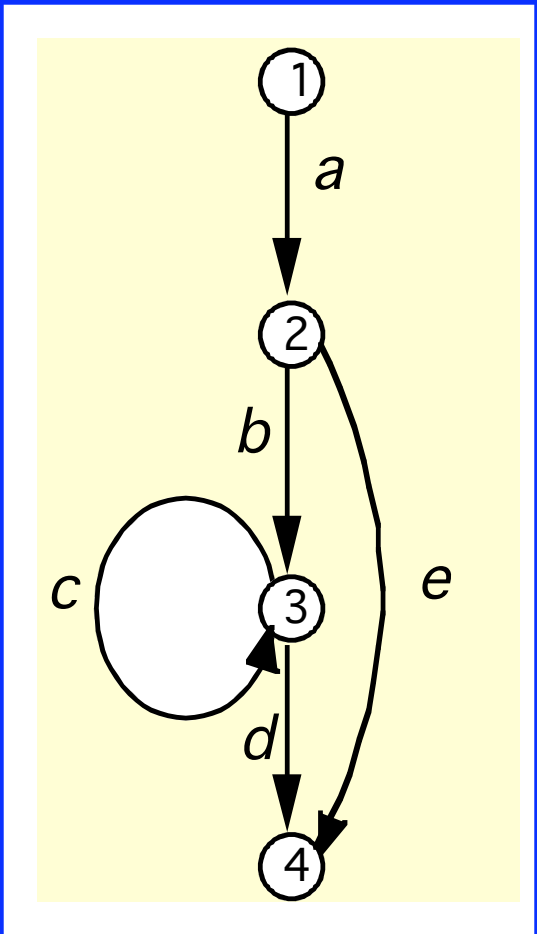


Extension

The same rules also apply to the cases where `a` and `b` are expressions describing complex path structures.

Hence a set of paths can be described by an expression composed of edge symbols and three connectives: concatenation, disjunction (+), and looping (*).

Example



The set of paths between nodes 1 and 4 in the graph shown above can be described by the regular expression

```
a(e+bc*d).
```

Loop

If p describes a path, then p* describes a loop formed by p and hence a set of paths obtained by iterating the loop for any number of times. Formally, p*= λ+p+pp+ppp+. . . . Here λ is a special symbol denoting the identity under concatenation (i.e., xλ = λx = x for any x) and is to be interpreted as a path of length zero (obtained by iterating the loop zero times).

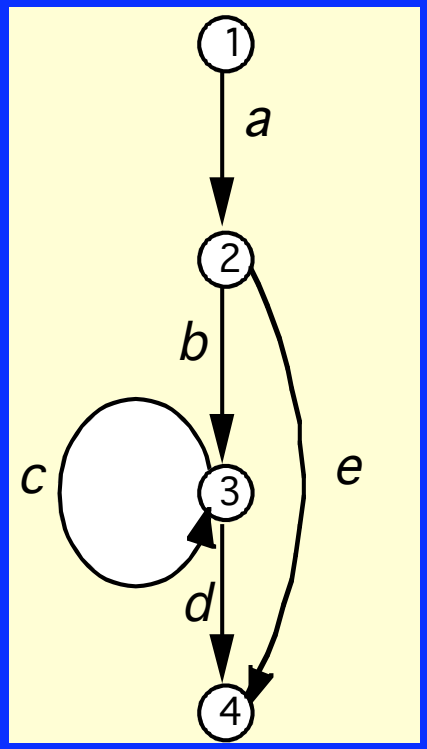
Matrix representation of a graph

Let G be a directed graph with n nodes. G can be represented by an n × n matrix as follows. First, the nodes in G are to be order in some way. Then we form an n × n matrix $[G] = [g_{ij}]$, where g_{ij} (the element on the i-th row and j-th column) is a regular expression denoting the set of all paths of length 1 (i.e., the paths formed by a single edge) leading from the i-th node to the j-th node.

Example

The graph given previously can be represented as

$$\begin{bmatrix} \emptyset & a & \emptyset & \emptyset \\ \emptyset & \emptyset & b & e \\ \emptyset & \emptyset & c & d \\ \emptyset & \emptyset & \emptyset & \emptyset \end{bmatrix}$$



∅: empty set.

Matrix operation

Let [X], [Y], [Z], and [W] be n × n matrices. We define

$$[X] + [Y] = [Z] = [z_{ij}]$$

where $z_{ij} = x_{ij} + y_{ij}$

Matrix operation (continued)

$[X][Y] = [W] = [w_{ij}]$

where $w_{ij} = \overset{n}{\underset{k=1}{+}} x_k y_{kj}$, and

$\overset{n}{\underset{k=1}{+}}$

$[X]^* = [X]^0 + [X]^1 + [X]^2 + [X]^3 + \ldots,$

where $[X]^0$ is defined to be an $n \times n$ matrix in which every element on the main diagonal is λ, and all other elements are identically $\varnothing$.

General form of a matrix

$$[G] = \begin{bmatrix} g_{11} & \cdots & g_{1k} & \cdots & g_{1n} \\ \cdots & & & & \cdots \\ g_{k1} & \cdots & g_{kk} & \cdots & g_{kn} \\ \cdots & & & & \cdots \\ g_{n1} & \cdots & g_{nk} & \cdots & g_{nn} \end{bmatrix}$$

Elimination of the k-th column and row

$$= \begin{bmatrix} g_{11} & \cdots & g_{1(k-1)} & g_{1(k+1)} & \cdots & g_{1n} \\ \cdots & & & & & \cdots \\ g_{(k-1)1} & \cdots & g_{(k-1)(k-1)} & g_{(k-1)(k+1)} & \cdots & g_{(k-1)n} \\ g_{(k+1)1} & \cdots & g_{(k+1)(k-1)} & g_{(k+1)(k+1)} & \cdots & g_{(k+1)n} \\ \cdots & & & & & \cdots \\ g_{n1} & \cdots & g_{n(k-)1} & g_{n(k+1)} & & g_{nn} \end{bmatrix}$$

Elimination of node k (continued)

$$+ \begin{bmatrix} g_{1k} \\ \cdots \\ g_{(k-1)k} \\ g_{(k+1)k} \\ \cdots \\ g_{nk} \end{bmatrix} \begin{bmatrix} g_{kk} \end{bmatrix} * \begin{bmatrix} g_{k1} & \cdots & g_{k(k-1)} & g_{k(k+1)} & \cdots & g_{kn} \end{bmatrix}$$

Elimination of node k (continued)

By repeatedly eliminating unessential nodes, we will be left with a 2×2 matrix shown below:

$$[B'] = \begin{bmatrix} b_{ii} & b_{ij} \\ b_{ji} & b_{jj} \end{bmatrix}$$

Elimination of node k (continued)

Then p_{ij} can be constructed from the elements in [B'] as follows:

$$p_{ij} = (b_{ii} + b_{ij}\, b_{jj}^{*} b_{ji})^{*} b_{ij} (b_{jj} + b_{ji}\, b_{ii}^{*} b_{ij})^{*}$$

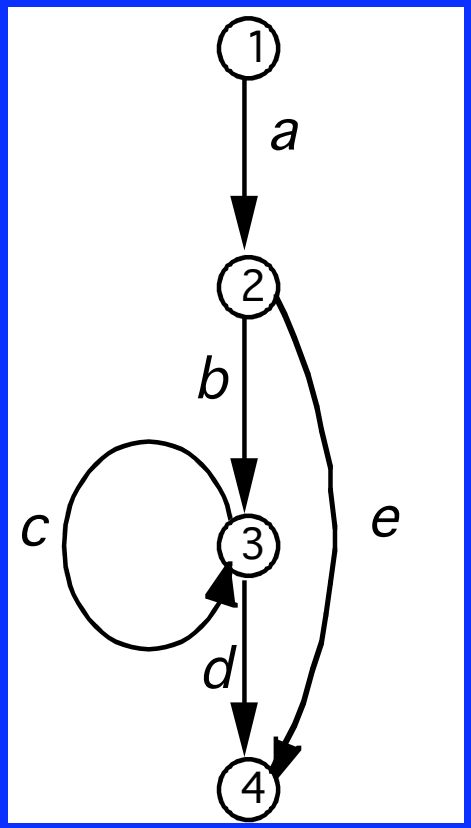
If $b_{ii} = b_{ji} = b_{jj} = \varnothing$, which is almost always the case in many applications, then we have

$$p_{ij} = b_{ij}$$

Example

Consider again the graph represented by

$$\begin{bmatrix} \emptyset & a & \emptyset & \emptyset \\ \emptyset & \emptyset & b & e \\ \emptyset & \emptyset & c & d \\ \emptyset & \emptyset & \emptyset & \emptyset \end{bmatrix}$$



Example (continued)

To find the set of all paths between nodes 1 and 4, we shall first eliminate node 2, i.e., column 2 and row 2 to yield

$$= \begin{bmatrix} \emptyset & \emptyset & \emptyset \\ \emptyset & c & d \\ \emptyset & \emptyset & \emptyset \end{bmatrix} + \begin{bmatrix} a \\ \emptyset \\ \emptyset \end{bmatrix} [\emptyset]^* \begin{bmatrix} \emptyset & b & e \end{bmatrix}$$

Example (continued)

$$= \begin{bmatrix} \emptyset & \emptyset & \emptyset \\ \emptyset & c & d \\ \emptyset & \emptyset & \emptyset \end{bmatrix} + \begin{bmatrix} \emptyset & ab & ae \\ \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix}$$

$$= \begin{bmatrix} \emptyset & ab & ae \\ \emptyset & c & d \\ \emptyset & \emptyset & \emptyset \end{bmatrix}$$

Example (continued)

Now column 2 and row 2 corresponds to the node labeled by integer 3. It can be similarly eliminated to yield the following 2×2 matrix.

Example (continued)

$$= \begin{bmatrix} \emptyset & ae \\ \emptyset & \emptyset \end{bmatrix} + \begin{bmatrix} ab \\ \emptyset \end{bmatrix} [c]^* \begin{bmatrix} \emptyset & d \end{bmatrix}$$

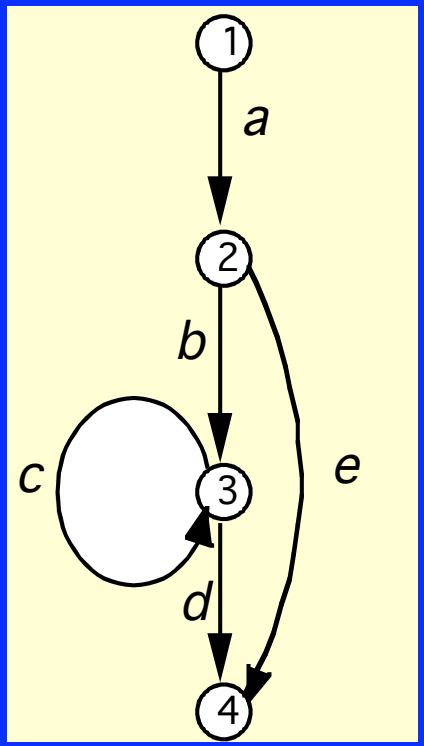
$$= \begin{bmatrix} \emptyset & ae \\ \emptyset & \emptyset \end{bmatrix} + \begin{bmatrix} \emptyset & abc^*d \\ \emptyset & \emptyset \end{bmatrix}$$

$$= \begin{bmatrix} \emptyset & ae + abc^*d \\ \emptyset & \emptyset \end{bmatrix}$$

Example (continued)

Hence, the set of paths leading from node 1 to node 4 is described by

ae + abc*d.



Remark

Generally speaking, the nodes can be eliminated in any order.

By eliminating the nodes in different order, the method may produce different regular expressions (with different complexities) as the result. But they all represent the same set of paths.