Chapter 17

Theory of Computation

Brooks/Cole
Thomson Learning™

OBJECTIVES

After reading this chapter, the reader should be able to:

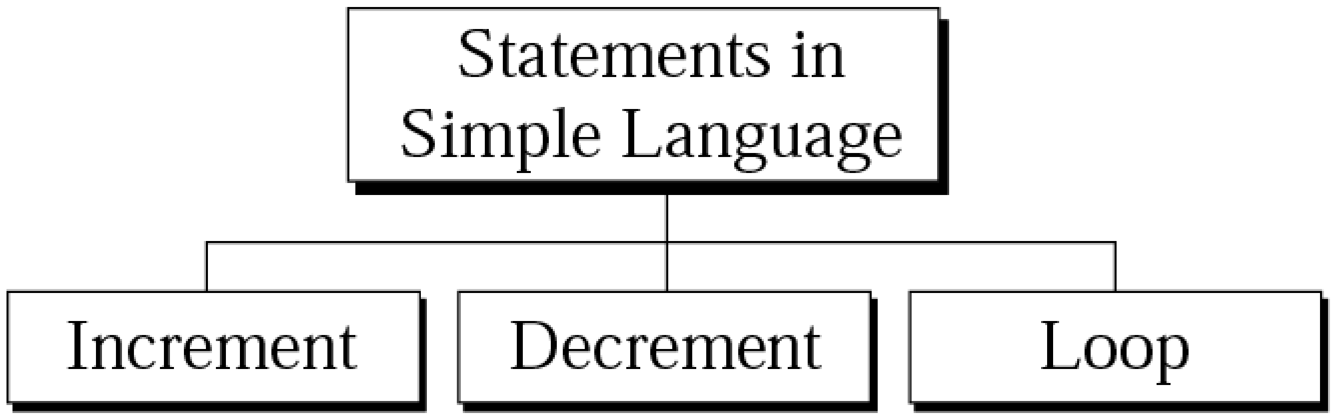
- Understand how a simple language with limited statements can solve any problem.
- Understand how the Turing machine can solve any problem that can be solved by a computer.
- Understand the Godel number and its importance in the theory of computation.
- Understand the ***halting problem*** as an example of a large set of problems that cannot be solved by a computer.

17.1

SIMPLE LANGUAGE

Figure 17-1

Statements in simple language



17.2

TURING MACHINE

Figure 17-2

Turing machine

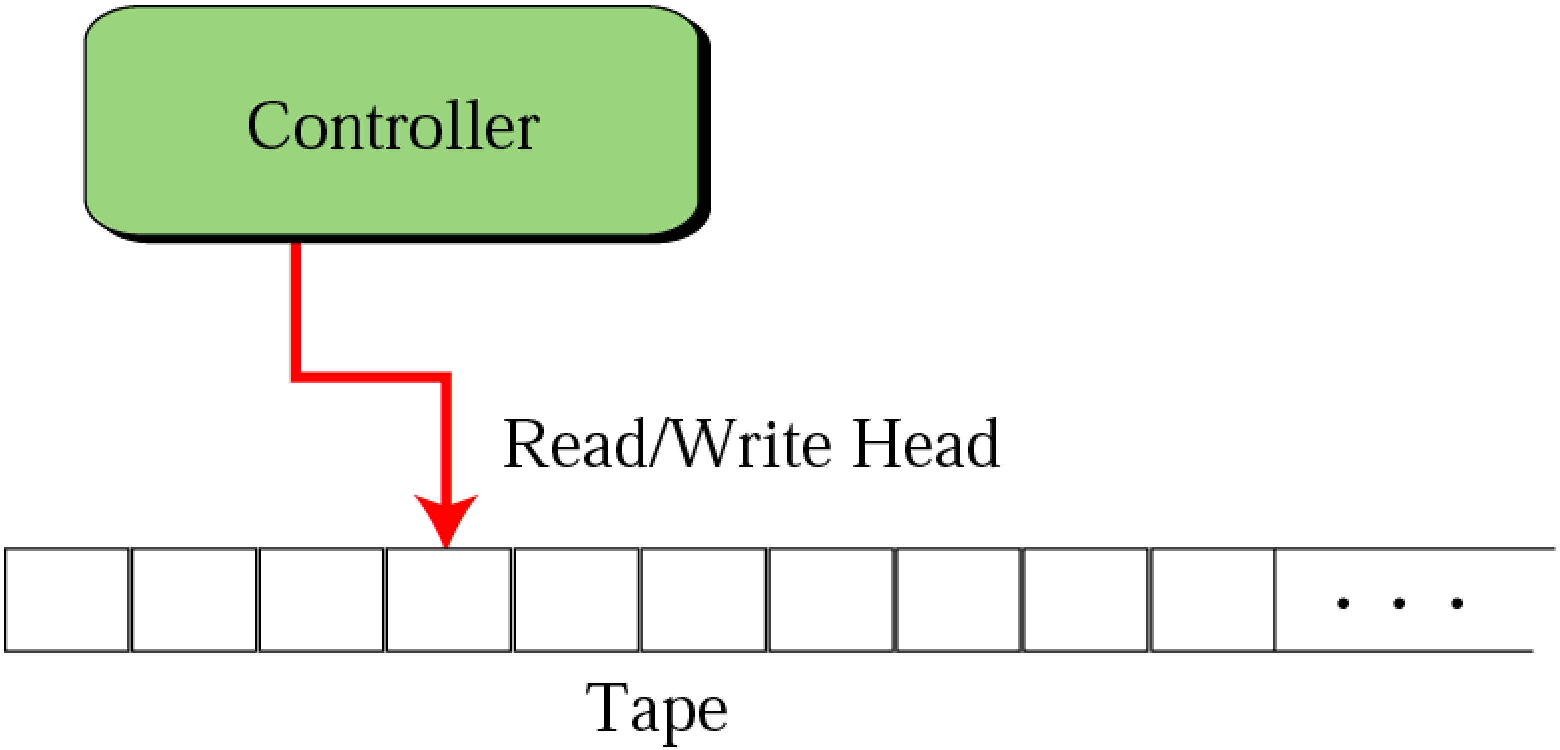


Figure 17-3

Tape

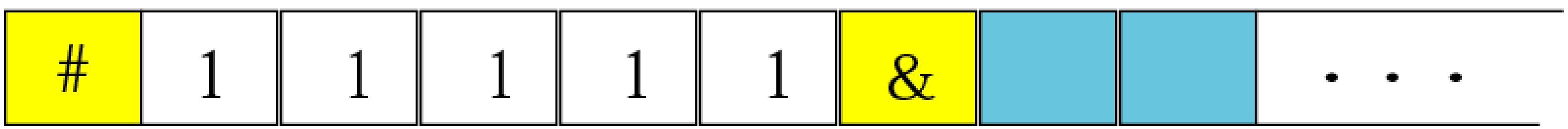


Figure 17-4

Transition state

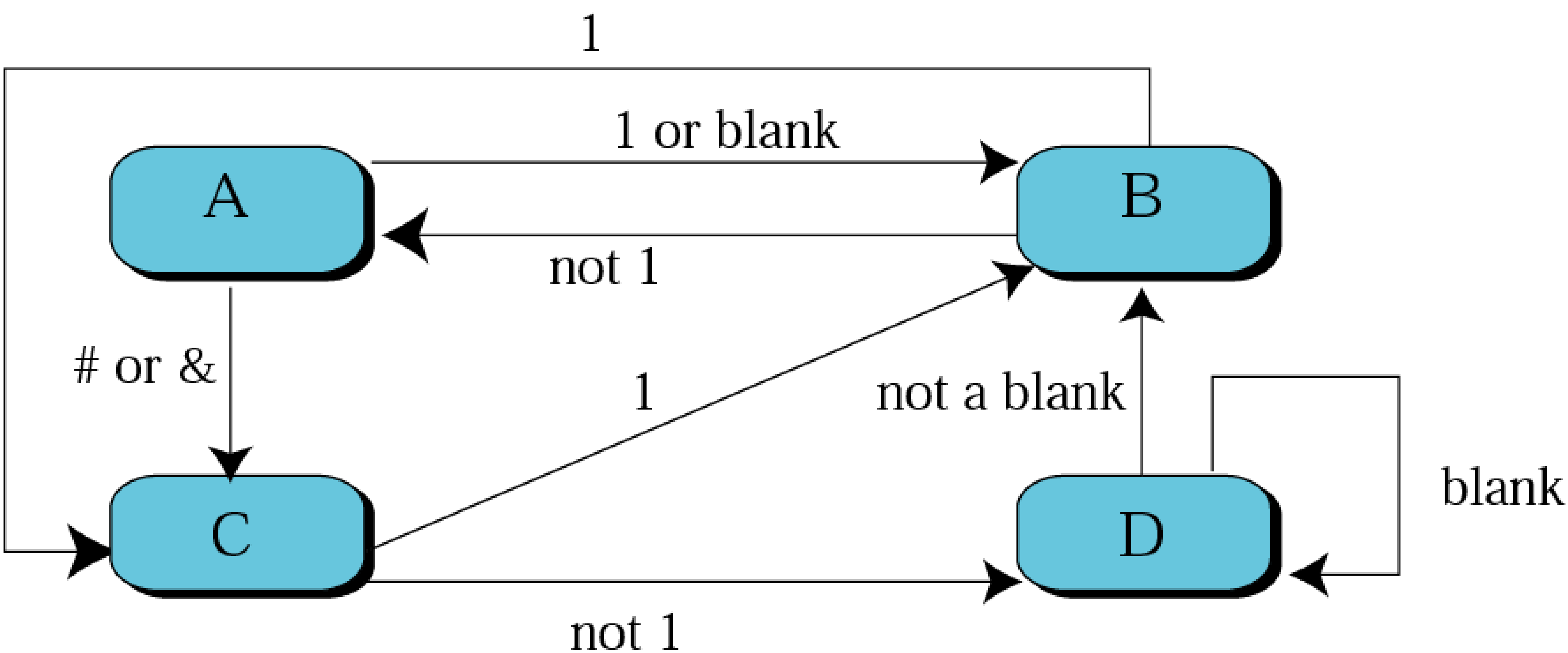


Table 17.1 Transitional table

Current State	***Read***	***Write***	***Move***	***New State***
A	1 or blank	#	→	B
A	# or &	&	←	C
B	1	1	←	C
B	not 1	same as read		A
C	1	blank	→	B
C	not 1	1	→	D
D	not a blank	same as read	→	B
D	blank	1	←	D

Figure 17-5

Transition diagram for incr x

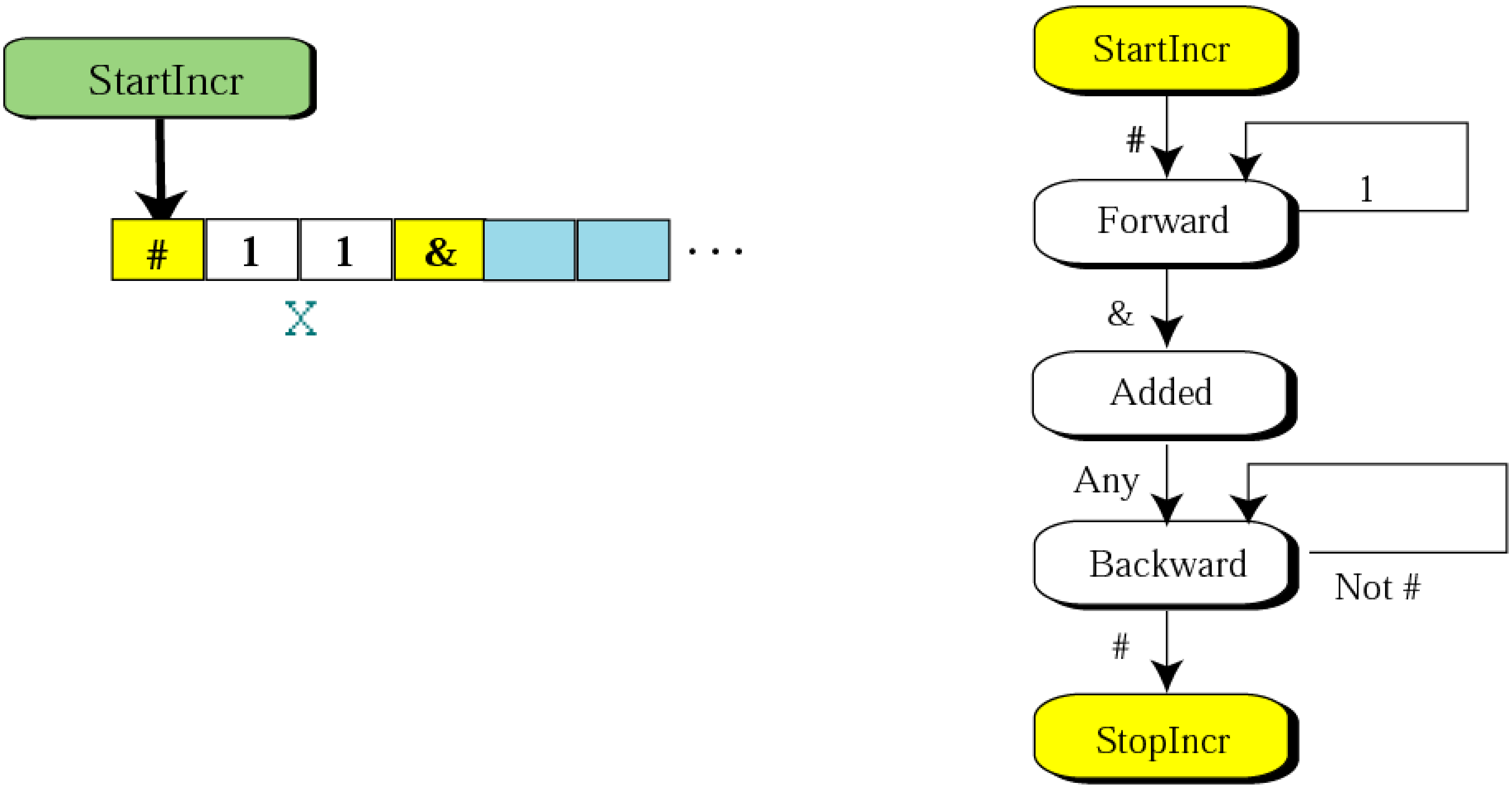
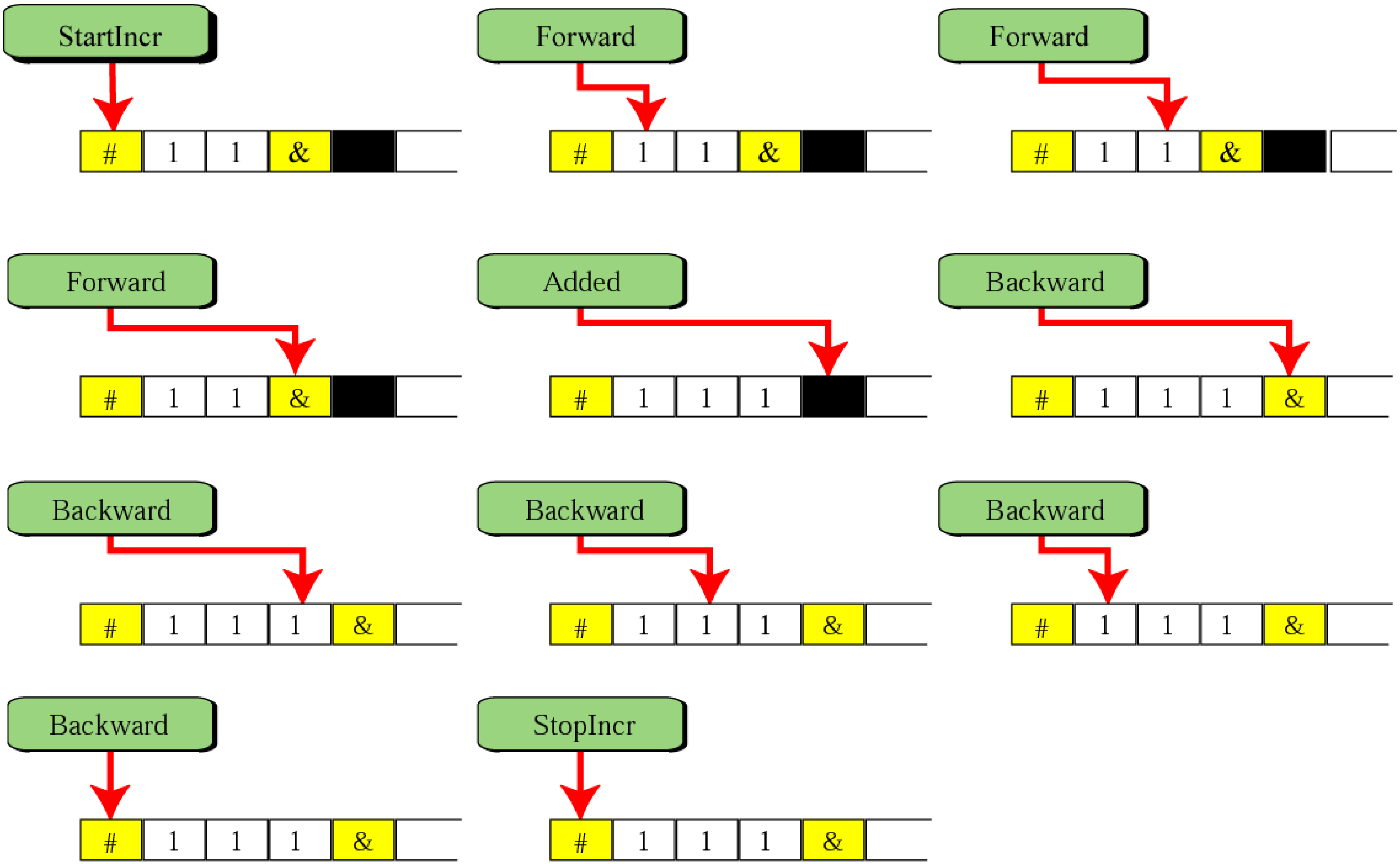


Table 17.2 Transitional table for incr x statement

Current State	Read	Write	Move	New State
StartIncr	#	#	→	Forward
Forward	1	1	→	Forward
Forward	&	1	→	Added
Added	any	&	←	Backward
Backward	not #	same as read	←	Backward
Backward	#	#		StopIncr

Figure 17-6

Steps in incr x statement



Brooks/Cole
Thomson Learning

Figure 17-7

Transition diagram for decr x

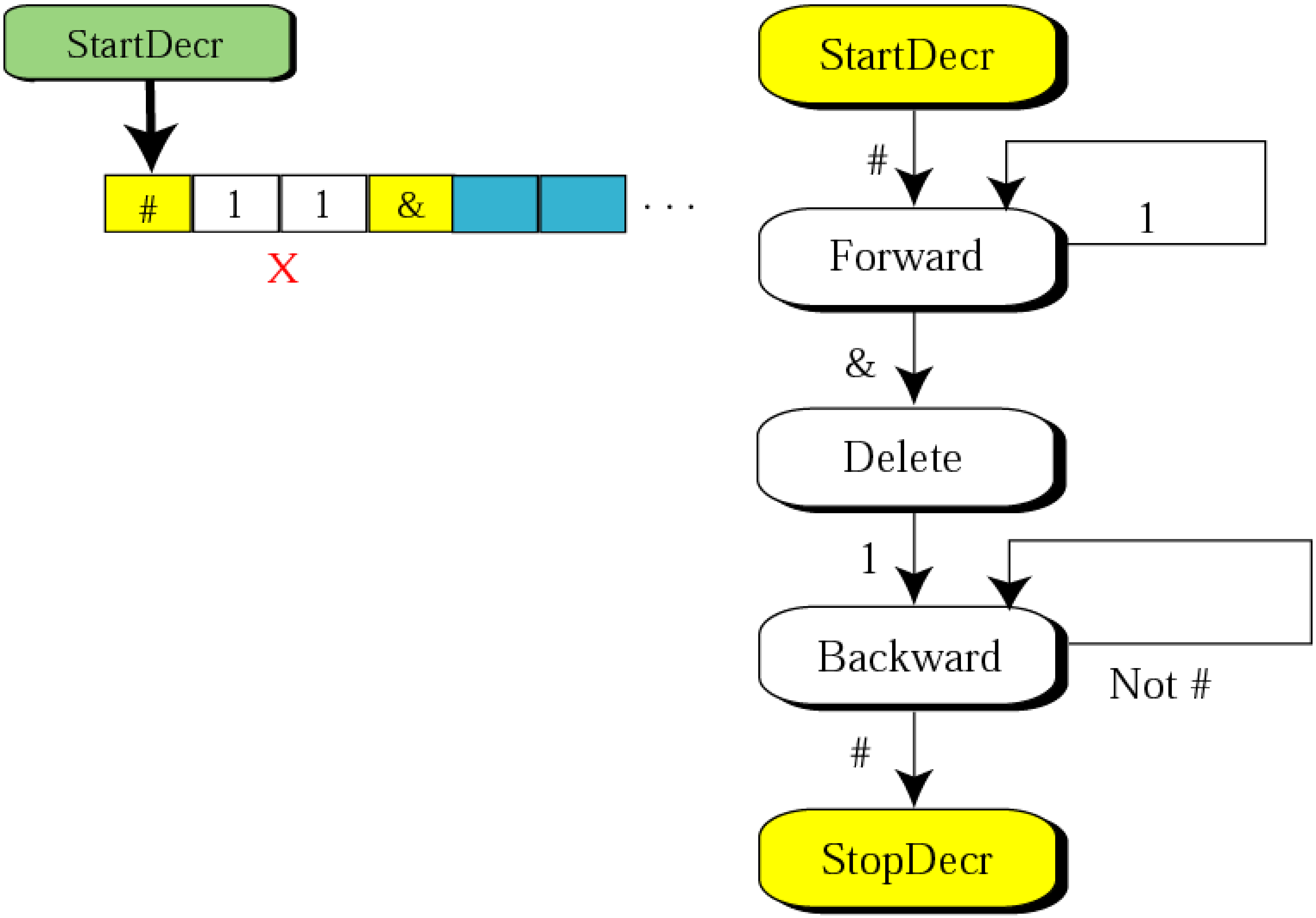
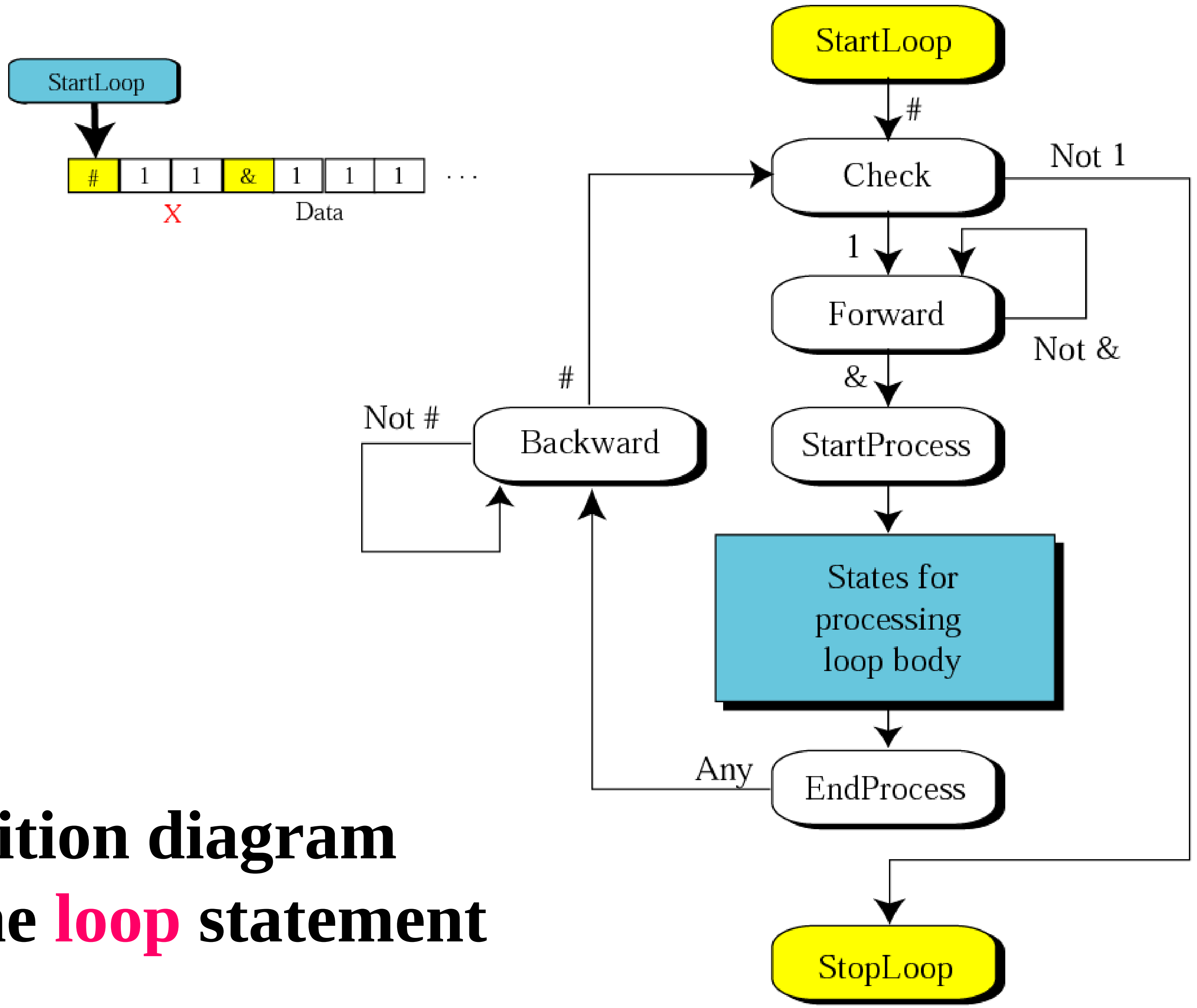


Table 17.3 Transitional table for decr x statement

Current State	Read	Write	Move	New State
StartDecr	#	#	→	Forward
Forward	1	1	→	Forward
Forward	&	blank	←	Delete
Delete	1	&	←	Backward
Backward	not #	same as read	←	Backward
Backward	#	#		StopDecr

Figure 17-8



Transition diagram for the loop statement

Table 17.4 Transitional table for the loop statement

Current State	Read	Write	Move	New State
---------	-------------	-------------	---------	---------
StartLoop	#	#	→	Check
Check	not 1	same as read	←	StopLoop
Check	1	1	→	Forward
Forward	not &	same as read	→	Forward
Forward	&	&	none	StartProcess
…	…	…	…	…
…	…	…	…	…
EndProcess	any	same as read	←	Backward
Backward	not #	same as read	←	Backward
Backward	#	#	none	Check

Brooks/Cole
Thomson Learning™

17.2

GODEL NUMBERS

Table 17.5 Code for symbols used in the Simple Language

Symbol	*Hex Code*	Symbol	*Hex Code*
1	1	9	9
2	2	incr	A
3	3	decr	B
4	4	while	C
5	5	{	D
6	6	}	E
7	7	x	F
8	8		

17.3

HALTING PROBLEM

A Classical Programming Question:

Can you write a program that tests whether or not any program, represented by its Godel number, will terminate?

Figure 17-9

Step 1 in proof

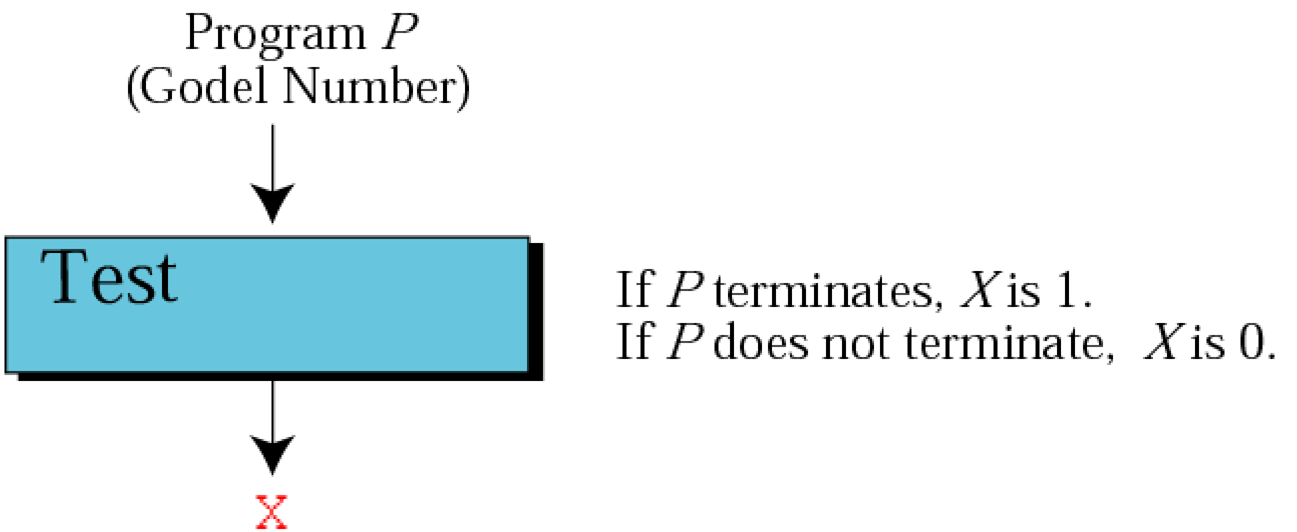
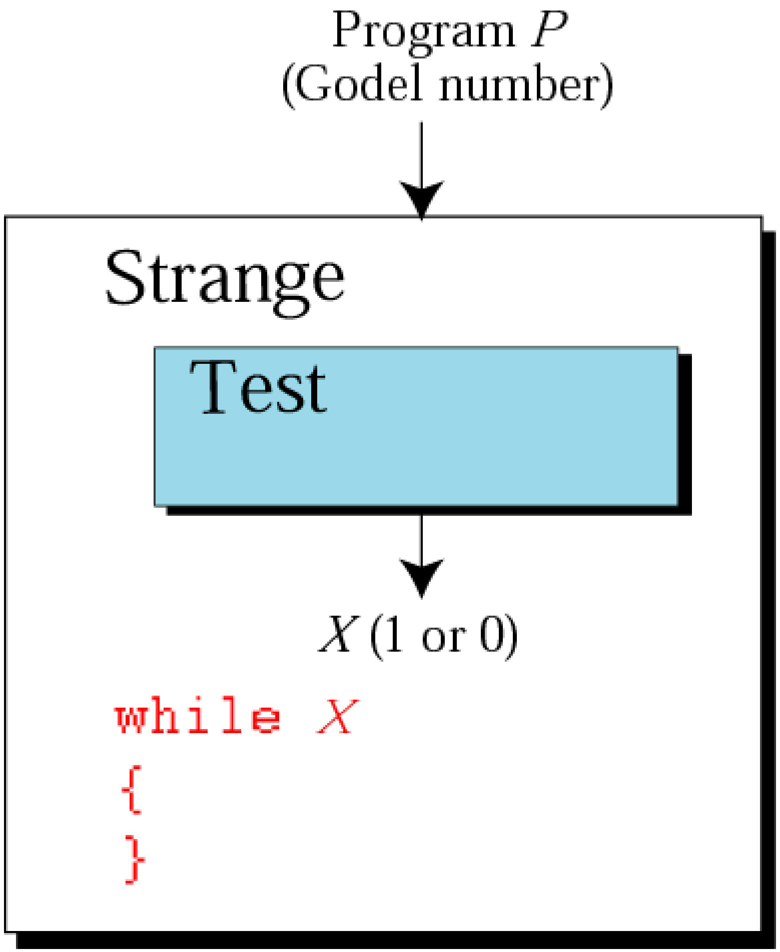


Figure 17-10

Step 2 in proof

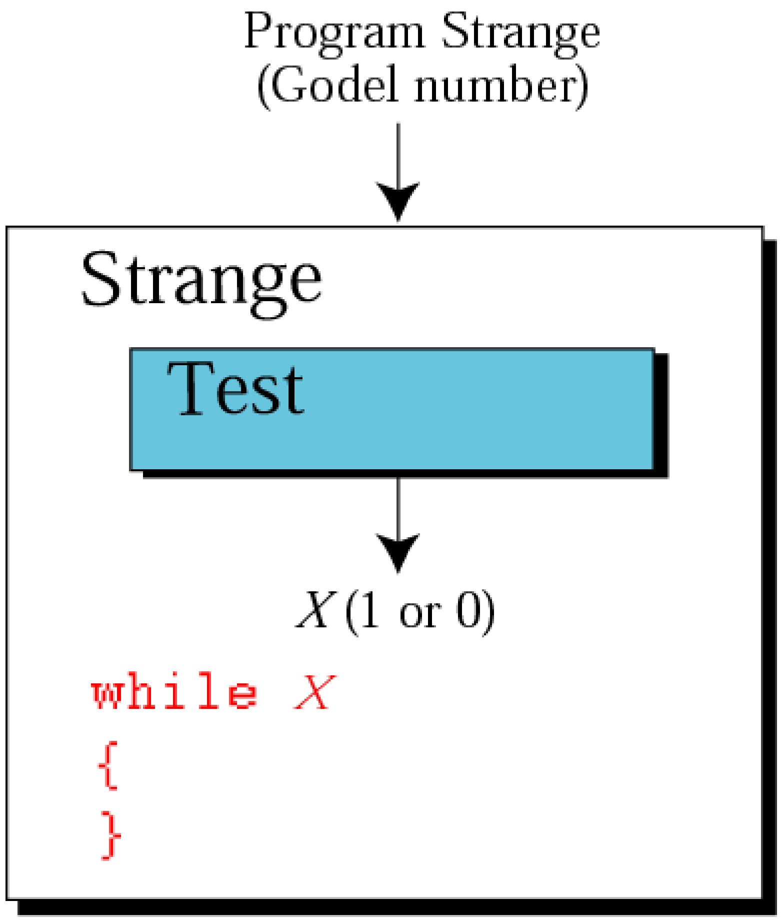


If P terminates, Strange does not terminate.

If P does not terminate, Strange terminates.

Figure 17-11

Step 3 in proof



If Strange terminates,
Strange does not terminate.

If Strange does not terminate,
Strange terminates.

17.5

SOLVABLE AND UNSOLVABLE PROBLEMS

Brooks/Cole
Thomson Learning™

Figure 17-12

Taxonomy of problems