



ບົດທີ 2

ການແກ້ປັນຫາດ້ວຍເຕັກນິກການຄົ້ນຫາ
(PROBLEM SOLVING USING SEARCH)

- ຂອບເຂດຂອງປັນຫາ (State Space)
- ການຄົ້ນຫາແບບງົມມືດ (Blind Search)
- ການຄົ້ນຫາແບບຮິວຣິສະຕິກ (Heuristic Search)
- MINIMAX ALGORITHM
- Alpha-Beta Cutoffs

2.1. ຂອບເຂດຂອງປັນຫາ (STATE SPACE)

ປັນຫາຜື້ນຖານໃນການພັດທະນາລະບົບປັນຍາປະດິດກໍຄືຜູ້ພັດທະນາຈະຕ້ອງເຂົ້າໃຈໃນເຕັກນິກຂອງການສ້າງໂປຣແກຣມ ເພື່ອຈະເຮັດໃຫ້ຄອມພິວເຕີສາມາດປະມວນຜົນໄດ້, ນັກວິໄຈທາງດ້ານປັນຍາປະດິດໃນຍຸກທຳອິດພະຍາຍາມຫາຄຳຕອບນີ້ ແລະ ຄົ້ນພົບວ່າລັກສະນະຢ່າງໜຶ່ງຂອງການນຳໃຊ້ໂປຣແກຣມຄອມພິວເຕີສາມາດເຮັດສິ່ງດັ່ງກ່າວໄດ້ກໍຄື **ລະບົບປັນຍາປະດິດຈະຕ້ອງມີຄວາມຮູ້ ແລະ ມີລະບົບການຫາຂໍ້ສະຫຼຸບຄວາມຮູ້ນັ້ນໄດ້**, ລະບົບນີ້ກໍຈະສາມາດແກ້ປັນຫາບາງຢ່າງໄດ້ຄືກັນກັບມະນຸດ

2.1. ຂອບເຂດຂອງປັນຫາ (STATE SPACE)

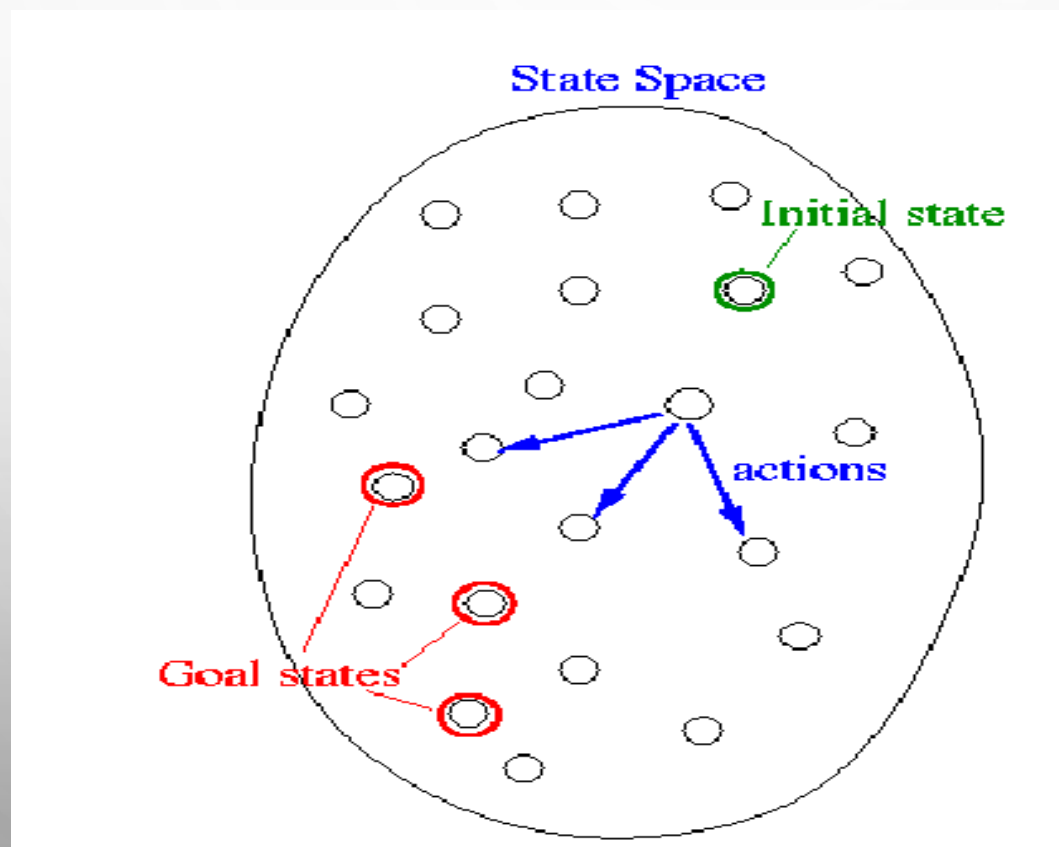
ເມື່ອປີ 1976 ເນວິວ ແລະ ໄຊມອນ (Newell and Simon, 1976) ໄດ້ສະເໜີແນວຄິດກ່ຽວກັບການເຮັດວຽກທີ່ເປັນອັດສະລິຍະວ່າສາມາດເກີດຂຶ້ນໄດ້ດ້ວຍຂະບວນການດັ່ງນີ້ :

- ການໃຊ້ຮູບແບບຂອງສັນຍາລັກ (Symbolic Pattern) ແທນຄຳອີງປະກອບທີ່ສຳຄັນຂອງປັນຫາໃນຂອບເຂດຂອງຄວາມຮູ້ນັ້ນ.
- ການປະມວນຜົນຮູບແບບຂອງສັນຍະລັກເຫຼົ່ານີ້ເພື່ອຫາຄຳຕອບທີ່ເປັນໄປໄດ້ (Possible Solution) ທັງໝົດອອກມາ.
- ຄົ້ນຫາເພື່ອເລືອກຄຳຕອບຈາກຄຳຕອບທີ່ເປັນໄປໄດ້ທັງໝົດທີ່ຖືກສ້າງຂຶ້ນມາໃນຂັ້ນຕອນກ່ອນໜ້ານັ້ນ.

2.1. ຂອບເຂດຂອງປັນຫາ (STATE SPACE)

ຂໍ້ສະຫຼຸບຂອງເນວິວ ແລະ ໄຊມອນ ເຮັດໃຫ້ເກີດການແກ້ປັນຫາແບບ
ຂອບເຂດສະຖານະ (State Space) ເຊິ່ງເປັນວິທີການແກ້ປັນຫາແບບ
ໜຶ່ງທີ່ກຳນົດວ່າປັນຫາມີຈຸດເລີ່ມຕົ້ນ ແລະ ເປົ້າໝາຍ ຂອງການແກ້ປັນຫາ,
ເຊິ່ງເອີ້ນວ່າສະຖານະຈຸດເລີ່ມຕົ້ນ ຫຼື ສະຖານະເລີ່ມຕົ້ນ (Initial State)
ເປັນສະຖານະທີ່ກຳນົດຈຸດເລີ່ມຕົ້ນຂອງການແກ້ປັນຫາ ຫຼື ໂຈດຂອງປັນຫາ
, ສຳລັບເປົ້າໝາຍ ຫຼື ສະຖານະເປົ້າໝາຍ (Gold State) ຄືສິ່ງທີ່ເຮົາຄາດ
ຫວັງຈະໄດ້ຈາກການແກ້ປັນຫານີ້ຄຳຕອບ.

2.1. ຂອບເຂດຂອງປັນຫາ (STATE SPACE)



2.1.1 ກຣາບ (GRAPH)

ຄືໂຄງສ້າງຂໍ້ມູນແບບບໍ່ເປັນເສັ້ນຊື່ (Non-Linear) ທີ່ປະກອບມີ ໂນດ (Node) ຫຼື ສະຖານະ ແລະ ເສັ້ນເຊື່ອມ (Edge ຫຼື Link), ເຊິ່ງໂນດໝາຍເຖິງສິ່ງທີ່ສາມາດກຳນົດຊື່ ແລະ ເກັບຂໍ້ມູນໄດ້, ສຳລັບເສັ້ນເຊື່ອມແມ່ນເສັ້ນທີ່ເຊື່ອມກັນລະວ່າງໂນດ

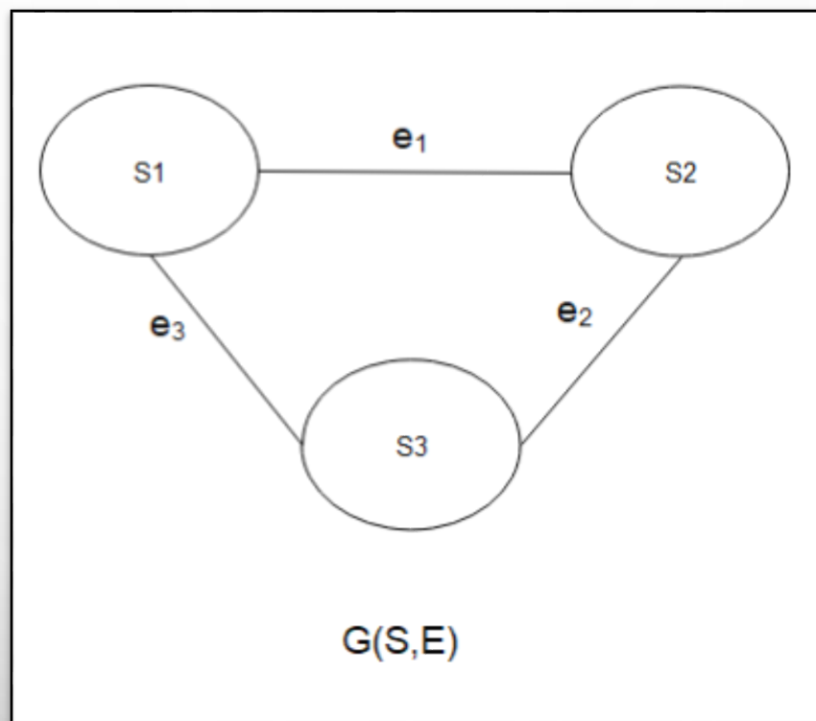
2.1.1 ກຣາບ (GRAPH)

ການກຳນົດທາງໄປຈາກໂນດໜຶ່ງໄປຫາອີກໂນດໜຶ່ງໂດຍທີ່ເສັ້ນທາງຈະບໍ່ຊ້ຳກັນ ໃນໂຄງສ້າງກຣາບຈະເອີ້ນວ່າ **ເສັ້ນທາງ (Path)**. ກຣາບໜຶ່ງຈະມີໂນດພິເສດເອີ້ນວ່າ **ຮາກ (Root)**, ເຊິ່ງຫມາຍເຖິງໂນດທີ່ມີເສັ້ນທາງຈາກໂຕມັນໄປຫາໂນດອື່ນຢູ່ເທິງກຣາບໄດ້ທຸກໂນດ, ຖ້າຫາກວ່າເສັ້ນທາງຈາກໂນດຮາກໄປຫາທຸກໂນດໃນກຣາບມີພຽງເສັ້ນທາງດຽວ ຈະເອີ້ນກຣາບນັ້ນວ່າ **ຕົ້ນໄມ້ (Tree)** ແຕ່ຖ້າມີເສັ້ນທາງຈາກຮາກໄປຍັງໂນດອື່ນຫຼາຍກ່ວາໜຶ່ງເສັ້ນທາງຈະເອີ້ນໂຄງສ້າງນີ້ວ່າ **ກຣາບ (Graph)**.

2.1.1 ກຣາບ (GRAPH)

ໂດຍທົ່ວໄປ ໃນໂຄງສ້າງຕົ້ນໄມ້ແລະກຣາບ ຈະຂຽນໃຫ້
ໂນດຮາກຢູ່ເທິງສຸດ ຖ້າມີໂນດຢູ່ເທິງຂຶ້ນໄປ 1 ໂນດ ໂນດ
ນັ້ນເອີ້ນວ່າ **ໂນດແມ່ (Parent Node)** ໃນທາງດຽວ
ກັນໂນດທີ່ຢູ່ລຸ່ມມັນເອີ້ນວ່າ **ໂນດລູກ (Child Node)**
ແລະ ໂນດທັງໝົດທີ່ແຕກອອກມາຈາກໂນດແມ່ດຽວກັນ
ເອີ້ນວ່າ **ໂນດຝີນ້ອງ (Sibling Node)**.

2.1.1 ກຣາບ (GRAPH)



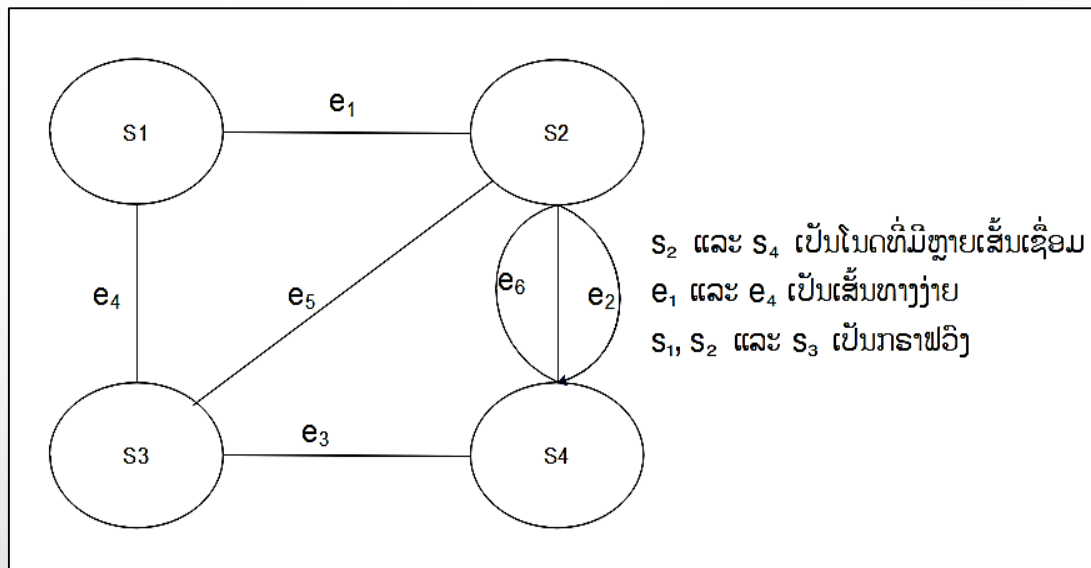
2.1.1 ກຣາບ (GRAPH)

ຖ້າກຳນົດສັນຍາລັກ G ແທນຄວາມໝາຍຂອງກຣາບ ອົງປະກອບທີ່ສຳຄັນຂອງ G ຈະປະກອບດ້ວຍກຸ່ມ ຂອງໂນດ ເຊິ່ງແທນດ້ວຍສັນຍາລັກ s ແລະ ກຸ່ມຂອງເສັ້ນເຊື່ອມ ເຊິ່ງແທນດ້ວຍສັນຍາລັກ E ແລະ ກຣາບ G ຈະແທນດ້ວຍ $G(S, E)$ ຖ້າ $\{s_1, s_2, s_3, \dots\}$ ຄືໂນດໃນກຸ່ມ s ແລະ $\{e_1, e_2, e_3, \dots\}$ ຄືເສັ້ນເຊື່ອມໃນກຸ່ມ e ເຮົາສາມາດຂຽນຄວາມສຳພັນຂອງເສັ້ນເຊື່ອມແລະໂນດໄດ້ດັ່ງນີ້ $e_1 = (s_1, s_2)$ ເຊິ່ງໝາຍເຖິງ e_1 ຄືເສັ້ນເຊື່ອມຕໍ່ລະຫວ່າງໂນດ s_1 ແລະ s_2 ແລະ ໂນດ s_1 ແລະ s_2 ຈະເອີ້ນວ່າ **ໂນດຂ້າງຄຽງ (Adjacent Node)**

2.1.1 ກຣາບ (GRAPH)

ກຣາບຈະຖືກເອີ້ນວ່າ *ກຣາບເຊື່ອມ (Connected Graph)* ຖ້າທຸກໂນດໃນກຣາບມີເສັ້ນທາງງ່າຍຕໍ່ເຖິງກັນໝົດ ແລະກຣາບຈະຖືກເອີ້ນວ່າ *ກຣາບສົມບູນ (Complete Graph)* ຖ້າທຸກໂນດໃນກຣາບເປັນໂນດຂ້າງຄຽງກັນໝົດ

2.1.1 ກຣາບ (GRAPH)



2.1.1 ກຣາບ (GRAPH)

- ຖ້າເສັ້ນເຊື່ອມໃນກຣາບມີການລະບຸຂໍ້ມູນຈະເອີ້ນວ່າ **ກຣາບປ້າຍ** (*Labelled Graph*) ແລະໃນກໍລະນີຂໍ້ມູນທີ່ລະບຸເປັນຕົວເລກທີ່ບອກຂະໜາດຂອງຄວາມສໍາຄັນຂອງເສັ້ນເຊື່ອມຈະເອີ້ນວ່າ **ນ້ຳໜັກ** (*Weight*) ໃນກໍລະນີທີ່ໂນດຄູ່ດຽວກັນ, ແຕ່ມີເສັ້ນເຊື່ອມຕໍ່ກັນຢູ່ຫຼາຍກວ່າໜຶ່ງເສັ້ນເຊື່ອມຈະເອີ້ນເສັ້ນເຊື່ອມເຫຼົ່ານັ້ນວ່າ **ເສັ້ນເຊື່ອມຫຼາຍ** (*Multiple Edges*) ເຊິ່ງເສັ້ນເຊື່ອມດັ່ງກ່າວສາມາດຂຽນໄດ້ດັ່ງນີ້:

$$e_1 (s_1, s_2) \text{ ແລະ } e_2 (s_1, s_2)$$

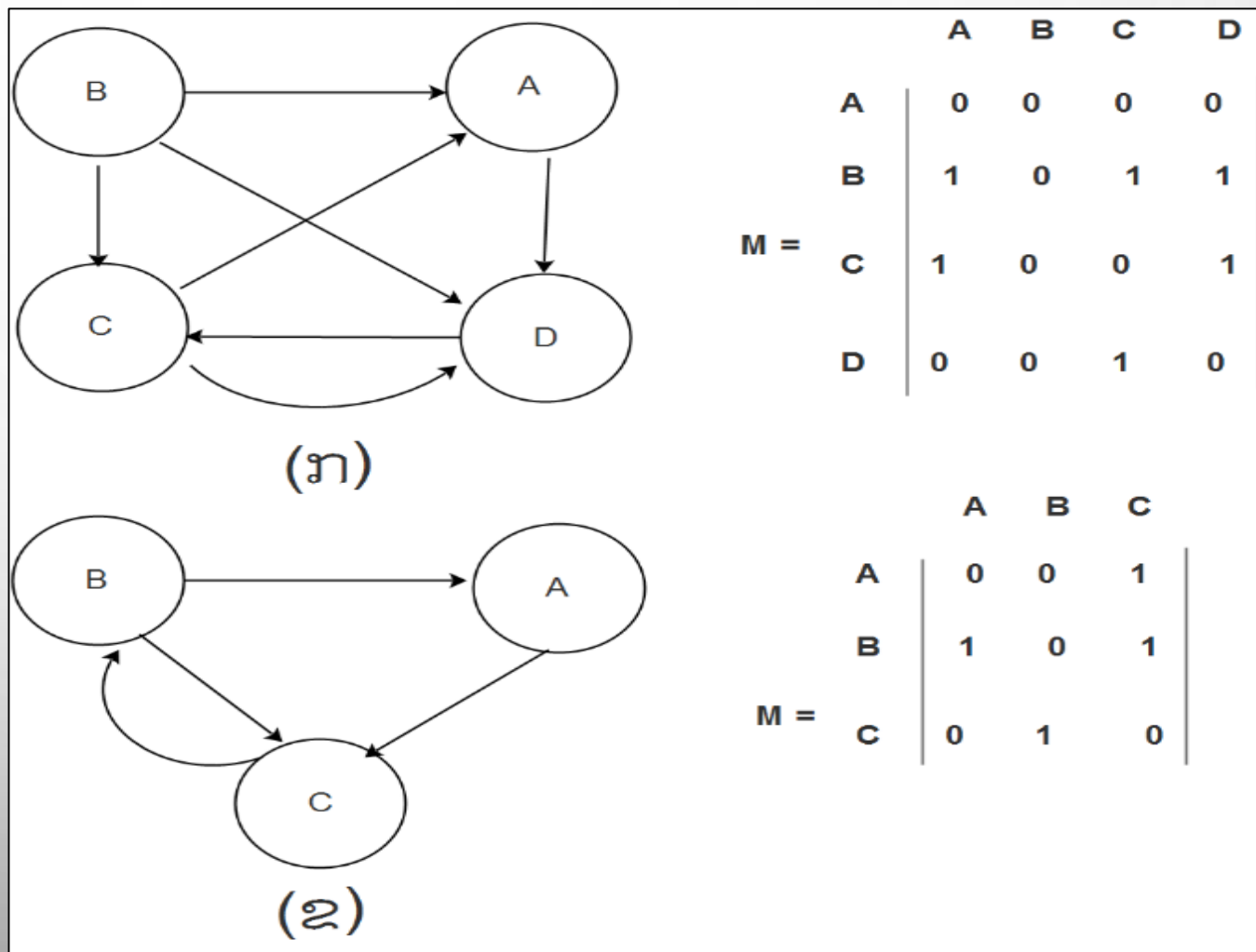
2.1.1 ກຣາບ (GRAPH)

ມາຕຣິດແບບໃກ້ກັນໃຊ້ແທນກຣາບ

$$M = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & \dots & e_j \\ \hline s_1 & a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ s_2 & a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ s_3 & a_{31} & a_{32} & a_{33} & \dots & a_{3j} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ s_i & a_{i1} & a_{i2} & a_{i3} & \dots & a_{ij} \end{array}$$

2.1.1 ກຣາບ (GRAPH)

ມາຕຣິດແບບໃກ້ກັນຂອງກຣາບ



2.1.2 ເຄື່ອງສະຖານະຈຳກັດ (FINITE STATE MACHINE)

ເຄື່ອງສະຖານະຈຳກັດ ຫຼື ເຄື່ອງອັດຕະໂນມັດຈຳກັດ (Finite Automation) ເປັນວິທີການອະທິບາຍຮູບແບບຂອງພຶດຕິກຳ ເຊັ່ນ: ການປິດ-ເປີດປະຕູລົບ ແລະ ການເຮັດວຽກຂອງໂຕແປງສ່ວນ (Parser) ເປັນຕົ້ນ, ທີ່ຂຽນແທນດ້ວຍກຣາບເຊື້ອມຕໍ່ຊະນິດກຣາບມິທິດ

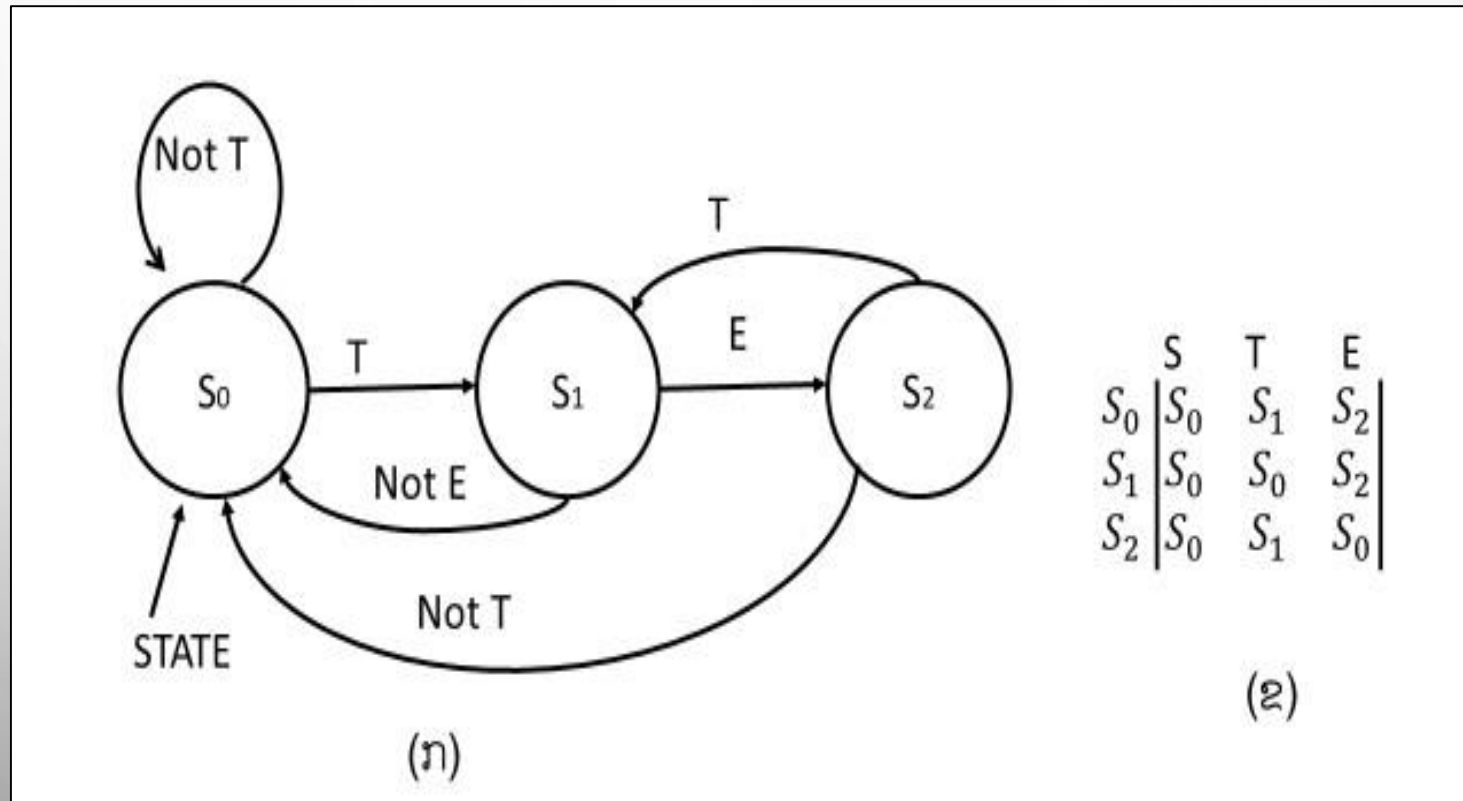
2.1.2 ເຄື່ອງສະຖານະຈຳກັດ (FINITE STATE MACHINE)

ເຄື່ອງສະຖານະຈຳກັດມີອົງປະກອບ 3 ຢ່າງຄື: ຊຸດຂອງ
ສະຖານະ ຫຼື ໂນດ, ຊຸດຂອງຄ່ານຳເຂົ້າ (Input Values)
ແລະ ຟັງຊັນການປ່ຽນສະຖານະ (State Transition
Function),

2.1.2 ເຄື່ອງສະຖານະຈຳກັດ (FINITE STATE MACHINE)

ເຄື່ອງສະຖານະຈຳກັດຊະນິດ *ຕົວຮັບ (Acceptor)* ເປັນເຄື່ອງສະຖານະຈຳກັດແບບງ່າຍຊະນິດໜຶ່ງ, ທີ່ຕອບຮັບ ຫຼື ປະຕິເສດຕໍ່ຄຳນຳເຂົ້າເທົ່ານັ້ນ, ຖ້າຄຳນຳເຂົ້າທັງໝົດທີ່ມີຕໍ່ສະຖານະປັດຈຸບັນໄດ້ຮັບການຕອບຮັບ, ຕຳນຳເຂົ້າກໍ່ຈະໄດ້ຮັບການຍອມຮັບ ແລະ ຈະປ່ຽນສະຖານະປັດຈຸບັນໄປຢູ່ສະຖານະຕໍ່ໄປ. ຖ້າສະຖານະປັດຈຸບັນຕອບປະຕິເສດ, ຄຳນຳເຂົ້ານັ້ນກໍ່ຈະຖືກປະຕິເສດຄືກັນ. ຕົວຢ່າງຂອງເຄື່ອງສະຖານະຈຳກັດຊະນິດຕົວຮັບ ທີ່ມີການຄົ້ນຫາຕົວອັກສອນໃນຄຳ, ຖ້າເຮົາຈະຫາຕົວອັກສອນວ່າ “TE” ຈາກຄຳ “STATE”

2.1.2 ເຄື່ອງສະຖານະຈຳກັດ (FINITE STATE MACHINE)

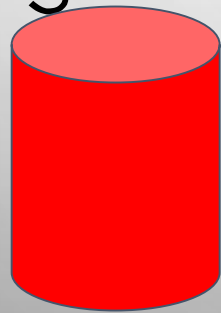


2.1.2 ເຄື່ອງສະຖານະຈຳກັດ (FINITE STATE MACHINE)

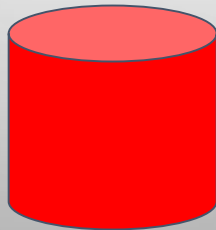
- ເຄື່ອງສະຖານະຈຳກັດຊະນິດຕົວຮັບ ສັນຍາລັກ 4 ຕົວຄື: (Σ, S, S_0, Δ) ເຊິ່ງ
- Σ ຄືຄຳນຳເຂົ້າ
- S ເປັນກຸ່ມຂອງສະຖານະ
- S_0 ເປັນສະຖານະຕ່າງໆໃນກຸ່ມຂອງ S
- Δ ເປັນຟັງຊັນການປ່ຽນສະຖານະເທົ່າກັບ $\Delta : S * \Sigma \rightarrow S$

2.1.3 ການກຳນົດນິຍາມໃຫ້ກັບປັນຫາ

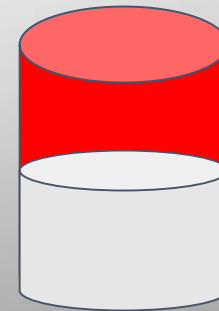
- ຄື ການອະທິບາຍລັກສະນະປັນຫາເພື່ອທີ່ຈະຫາວິທີການແກ້ໄຂ
- ຕົວຢ່າງທີ່ 1: ປັນຫາໂທນ້ຳ ກຳນົດໃຫ້ມີໂທນ້ຳ 2 ອັນ ອັນ 1 ມີຄວາມຈູ 4 ລິດ ແລະອັນທີ່ສອງມີຄວາມຈູ 3 ລິດ ຈະເຮັດຢ່າງໃດໃຫ້ໂທນ້ຳອັນທຳອິດນີ້ມີຄວາມຈູ 2 ລິດພໍດີ ໂດຍບໍ່ມີເຄື່ອງວັດໃດ ໆ



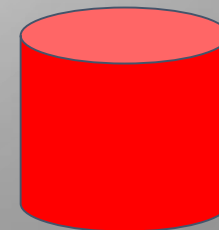
4 ລິດ



3 ລິດ



ປັນຈຸນ້ຳ 2 ລິດ



2.1.3 ການກຳນົດນິຍາມໃຫ້ກັບປັນຫາ

1. ຄ່າຂອງໂອກາດຕ່າງ ໆ ທີ່ຈະເກີດຂຶ້ນໃນແຕ່ລະສະຖານະ ກຳນົດໃຫ້ ຕົວປ່ຽນ x ແທນປະລິມານນ້ຳໃນໂທນ້ຳທີ່ 1 ແລະ y ແທນປະລິມານນ້ຳໃນໂທທີ່ສອງ

ຈະໄດ້ວ່າ $x = 0, 1, 2, 3, 4$ (ທຳອິດຄວາມຈຸ 4 ລິດ)

$y = 0, 1, 2, 3$ (ທຳອິດຄວາມຈຸ 3 ລິດ)

2. ກຳນົດຈຸດເລີ່ມຕົ້ນແລະສິ້ນສຸດການແກ້ປັນຫາ

- ສະຖານະເລີ່ມຕົ້ນ ກຳນົດໃຫ້ສະຖານະເລີ່ມຕົ້ນຂອງປັນຫານີ້ຄື ບໍ່ມີນ້ຳໃນໂທນ້ຳທັງສອງ ດັ່ງນັ້ນເຮົາຈຶ່ງໃຊ້ຄູ່ອັນດັບ (x, y) $x =$ ປະລິມານນ້ຳໃນໂທນ້ຳທຳອິດ ແລະ $y = n$ ປະລິມານນ້ຳໃນໂທທີ່ສອງ

- ສະຖານະສິ້ນສຸດ ຕ້ອງການໃຫ້ເຫລືອນ້ຳໃນໂທນ້ຳທຳອິດພຽງ 2 ລິດ ສ່ວນໂທນ້ຳໃບທີ່ສອງຈະເຫລືອຄວາມຈຸເທົ່າໃດກໍໄດ້

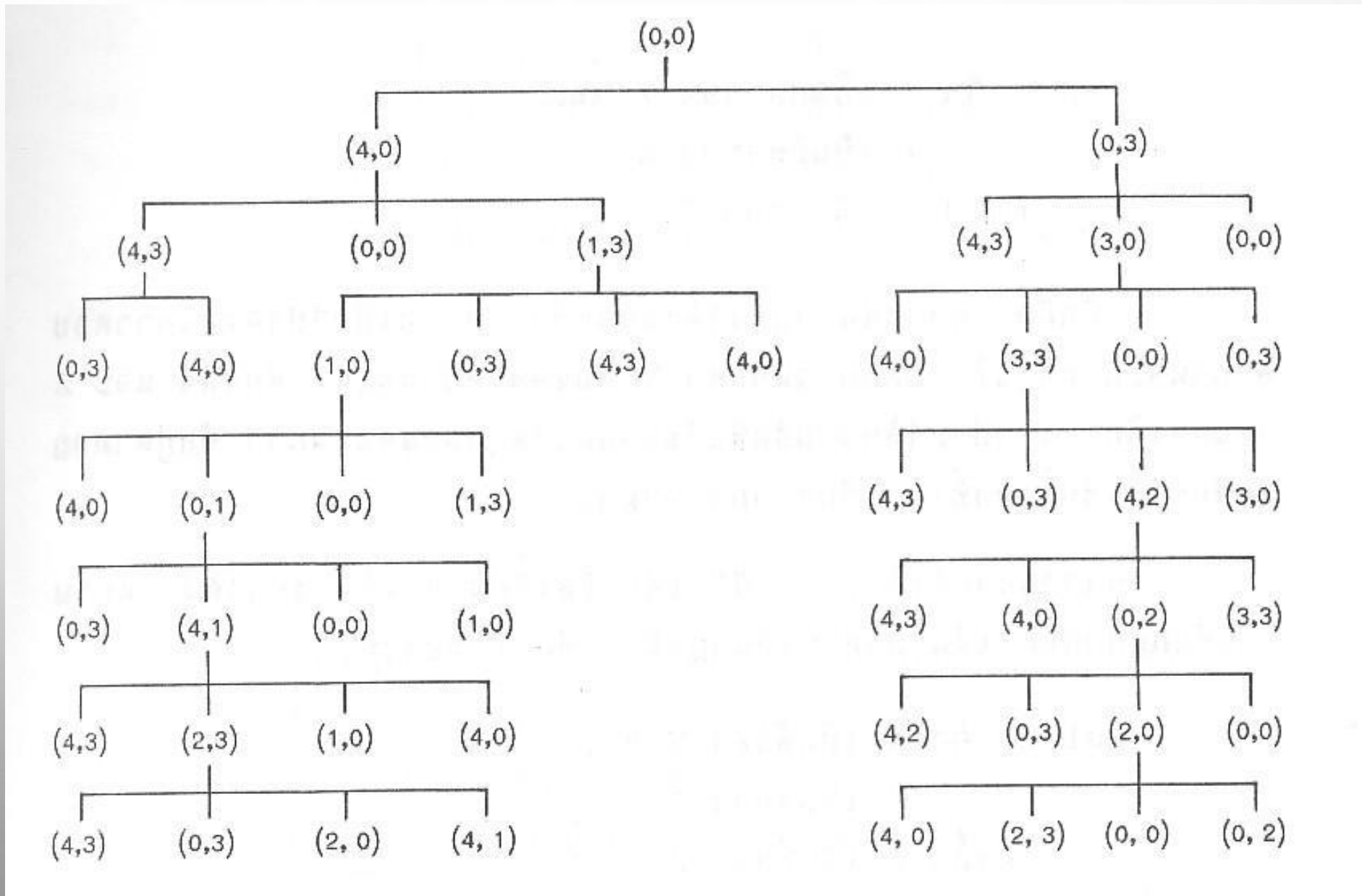
2.1.3 ການກຳນົດນິຍາມໃຫ້ກັບປັນຫາ

3. ກົດທີ່ອະທິບາຍປັນຫານັ້ນ ໆ

- ໃສ່ນ້ຳໃນໂທ 1 ເຕັມ
- ໃສ່ນ້ຳໃນໂທໃບທີ່ສອງຈົນເຕັມ
- ເທນ້ຳບາງສ່ວນອອກຈາກໂທ 1
- ເທນ້ຳບາງສ່ວນອອກຈາກໂທໃບທີ່ສອງ

ປ່ຽນກົດໃຫ້ເປັນເງື່ອນໄຂທີ່ງ່າຍຕໍ່ການນຳໄປໃຊ້ຈະໄດ້ດັ່ງນີ້

ລຳດັບ	ກົດເກນ	ຄຳອະທິບາຍ
1	$(X,Y: X < 4) \rightarrow (4,Y)$	ຖອກນ້ຳໃສ່ໂຕທຳອິດຈົນເຕັມ
2	$(X,Y: Y < 3) \rightarrow (X,3)$	ຖອກນ້ຳໃສ່ໂຕທີ 2 ຈົນເຕັມ
3	$(X,Y: X > D) \rightarrow (X-D,Y)$	ຖອກນ້ຳບາງສ່ວນອອກຈາກໂຕ 2
4	$(X,Y: Y > D) \rightarrow (X,Y-D)$	ຖອກນ້ຳອອກຈາກໂຕທຳອິດຈົນໝົດ
5	$(X,Y: Y > 0) \rightarrow (X,0)$	ຖອກນ້ຳອອກຈາກໂຕທີ 2 ຈົນໝົດ
6	$(X,Y: X+Y \geq 4 \wedge Y > 0) \rightarrow (4, Y-(4-X))$	ຖອກນ້ຳອອກຈາກໂຕທີ 2 ໃສ່ໂຕທຳອິດຈົນເຕັມ ໂດຍໂຕທີ 1 ມີນ້ຳລວມກັນຫຼາຍກວ່າ 4 ລິດ
7	$(X,Y: X+Y \geq 3 \wedge X > 0) \rightarrow (X-(3-Y),3)$	ຖອກນ້ຳອອກຈາກໂຕທຳອິດ ໃສ່ໂຕທີ 2 ຈົນເຕັມ ໂດຍໂຕທີ 2 ມີນ້ຳລວມກັນຫຼາຍກວ່າ 3 ລິດ
8	$(X,Y: X+Y \leq 4 \wedge Y > 0) \rightarrow (X+Y,0)$	ຖອກນ້ຳອອກຈາກໂຕທີ 2 ໃສ່ໂຕທຳອິດຈົນເຕັມ ໂດຍໂຕທີ 1 ມີນ້ຳລວມກັນຫຼາຍກວ່າ 4 ລິດ
9	$(X,Y: X+Y \leq 3 \wedge X > 0) \rightarrow (0,X+Y)$	ຖອກນ້ຳອອກຈາກໂຕທີ 2 ໃສ່ໂຕທີ 1
10	$(X > 0, X+Y \leq 3) \rightarrow (0, X+Y)$	ຖອກນ້ຳອອກຈາກໂຕທຳອິດ ໃສ່ໂຕທີ 2



ໂຖ 1	ໂຖ 2	ກົດຂໍ້ທີ	ສະຖານະ
0	0		ເລີ່ມຕົ້ນ
0	3	2	
3	0	7	
3	3	2	
4	2	5	
0	2	3	
2	0	7	ເປົ້າໝາຍ

2.1.3 ການກຳນົດນິຍາມໃຫ້ກັບປັນຫາ

- ການແກ້ປັນຫາແບບທີ່ກ່າວມາແລ້ວເປັນຕົວຢ່າງຂອງການແກ້ປັນຫາແບບ ລະບົບການຜະລິດ (Production) ເຊິ່ງເປັນຂະບວນການໃນການວາງໂຄງສ້າງໂປຣແກຣມແບບປັນຍາປະດິດ ໃຫ້ມີລັກສະນະງ່າຍຕໍ່ການອະທິບາຍ ຂະບວນການ (ຫຼື ວິທີການແກ້ປັນຫາ) ການແກ້ປັນຫາແບບນີ້ຈະຕ້ອງປະກອບໄປດ້ວຍ

2.1.3 ການກຳນົດນິຍາມໃຫ້ກັບປັນຫາ

1. ກົດເຊິ່ງກົດແຕ່ລະຂໍ້ຈະຕ້ອງປະກອບດ້ວຍສ່ວນທີ່ຢູ່ທາງຊ້າຍທີ່ຈະອະທິບາຍເຖິງເງື່ອນໄຂຂອງກົດ ແລະ ສ່ວນທີ່ຢູ່ທາງຂວາທີ່ອະທິບາຍເຖິງຜົນຂອງກົດ
2. ຖານຂໍ້ມູນທີ່ມີການສະຫຼຸບທີ່ຕ້ອງການ ບາງສ່ວນຂອງຖານຂໍ້ມູນຈະເປັນແບບຖາວອນ ແລະບາງສ່ວນຈະເປັນສິ່ງທີ່ກ່ຽວຂ້ອງກັບການແກ້ປັນຫາໃນຊ່ວງນັ້ນ
3. ກົນໄກໃນການຄວບຄຸມ ເປັນສ່ວນທີ່ກຳນົດລຳດັບຂອງກົດທີ່ຈະນຳມາໃຊ້ຫຼືປຽບທຽບກັບຖານຂໍ້ມູນເພື່ອບອກລຳດັບຂອງການແກ້ປັນຫາ ສຳລັບກົນໄກໃນການຄວບຄຸມນີ້ມີຢູ່ 3 ສິ່ງທີ່ສຳຄັນຄື:

ການກຳນົດທິດທາງສຳລັບການຄົ້ນຫາ

ຂະບວນການໃນການເລືອກກົດເກນ ແລະ

ການຄົ້ນຫາແບບຮິວຣີສະຕິກ

2.1.4 ວິທີການຕ່າງໆ ໃນການແກ້ປັນຫາ

ການແກ້ປັນຫາຫາຂອງປັນຍາປະດິດນັ້ນ ມີເລື່ອງໃຫຍ່ທີ່ຈະຕ້ອງ
ພິຈາລະນາເຖິງຄື:

1. ການກຳນົດທິດທາງສຳລັບການຄົ້ນຫາ ແລະ ຮູບແບບຂອງໂຄງ
ສ້າງຂໍ້ມູນທີ່ໃຊ້ສຳລັບການຄົ້ນຫາ
2. ການສະແດງຄວາມຮູ້
3. ຂະບວນການໃນການເລືອກກົດເກນ ແລະ
4. ການຄົ້ນຫາແບບຮິວຣິສະຕິກການກຳນົດທິດທາງສຳລັບການ
ຄົ້ນຫາ ແລະ ຮູບແບບຂອງໂຄງສ້າງຂໍ້ມູນທີ່ໃຊ້ສຳລັບການຄົ້ນ
ຫາ

2.1.5 ການສະແດງຄວາມຮູ້

ເປັນຂະບວນການໃນການສະແດງຄວາມຫມາຍທີ່ປາກົດຢູ່ໃນແຕ່ລະໂນດວ່າຈະມີ
ວິທີການແນວໃດ

ຄວາມຮູ້ແລະຄວາມເຂົ້າໃຈທີ່ເກີດຂຶ້ນຂອງບຸກຄົນໜຶ່ງ ໆ ຈະຕ້ອງມີການຖ່າຍທອດ
ໃຫ້ເປັນຮູບແບບສັນຍະລັກເພື່ອໃຫ້ຄົນອື່ນ ໆ ໄດ້ເຂົ້າໃຈ ເຊັ່ນກັນໃນການແກ້
ບັນຫາດ້ວຍຄອມພິວເຕີ ເຮົາຈຳເປັນຕ້ອງສະແດງຄວາມຮູ້ໃຫ້ຢູ່ໃນຮູບທີ່ສາມາດສື່
ໃຫ້ມະນຸດເຂົ້າໃຈໄດ້ ແລະຄອມພິວເຕີກໍເຂົ້າໃຈໄດ້ເຊັ່ນກັນ ສຳລັບລາຍລະອຽດ
ຂອງວິທີການສະແດງຄວາມຮູ້ຈະໄດ້ຮຽນໃນຫົວຂໍ້ການສະແດງຄວາມຮູ້ອີກເທື່ອໜຶ່ງ

2.1.5 ການສະແດງຄວາມຮູ້

ໃນກໍລະນີທີ່ຄວາມຮູ້ມີຄວາມຊັບຊ້ອນຂຶ້ນມີເລື່ອງຫຼາຍຢ່າງທີ່ຈະຕ້ອງ
ພິຈາລະນາ ເຊັ່ນ: ໃນກໍລະນີທີ່ຄວາມຮູ້ບໍ່ແມ່ນຕົວເລກ ແຕ່ແມ່ນອອບເຈັກ
(object) ແລະ ຄວາມຈິງ (fact) ທີ່ມີຄວາມສໍາພັນກັນ ເຊັ່ນ: ຄວາມຮູ້
'Plant is on the table' ຈະມີອອບເຈັກ 2 ຕົວຄື Plant ແລະ
Table ທີ່ມີ on ສະແດງເຖິງຄວາມສໍາພັນ ການສະແດງຄວາມຮູ້ແບບນີ້ຈະ
ມີການກ່າວເຖິງຕໍ່ໄປຢ່າງລະອຽດ ສໍາລັບຕົວຢ່າງຂອງການສະແດງຄວາມຮູ້
ແບບນີ້ມີຕົວຢ່າງດັ່ງຕໍ່ໄປນີ້:

2.1.5 ການສະແດງຄວາມຮູ້

- ON(plant,table) : plant is on the table
- IN(table,room) : table is in the room
- UNDER(table>window) : table is under the window

2.1.5 ການສະແດງຄວາມຮູ້

ຢ່າງໃດກໍຕາມການສະແດງຄວາມຮູ້ມີສິ່ງທີ່ຄວນຈະຄໍານຶງເຖິງດັ່ງຕໍ່ໄປນີ້:

1. ຄວາມຮູ້ທັງໝົດຈະສາມາດລວມເປັນຄວາມຮູ້ດຽວກັນໄດ້ແນວໃດ ເຊັ່ນຫາກວ່າເຮົາກໍາລັງອະທິບາຍເຖິງລັກສະນະຂອງຫ້ອງຫ້ອງໜຶ່ງທີ່ບອກວ່າ ຫ້ອງນີ້ຕັ້ງໂຕະໄວ້ກ້ອງປ້ອງຢ້ຽມ 'table is under the window' ແລ້ວວມີໜຶ່ງເມື່ອມີການປ່ຽນຖານຄວາມຮູ້ວ່າ `CENTER(table,room)` ໃນລະບົບການສະແດງຄວາມຮູ້ຈະມີວິທີການແນວໃດທີ່ຈະເຮັດໃຫ້ຮູ້ວ່າ `UNDER(table>window)` ໃຊ້ບໍ່ໄດ້ແລ້ວເພາະເມື່ອໂຕະມາຢູ່ກາງຫ້ອງກໍເປັນໄປບໍ່ໄດ້ທີ່ໂຕະໂຕດຽວກັນຈະຢູ່ກ້ອງປ້ອງຢ້ຽມ.

2.1.5 ການສະແດງຄວາມຮູ້

ຢ່າງໃດກໍຕາມການສະແດງຄວາມຮູ້ມີສິ່ງທີ່ຄວນຈະຄໍານຶງເຖິງດັ່ງຕໍ່ໄປນີ້:

2. ການຈັດລຳດັບແນວໃດເຮັດໃຫ້ການຄົ້ນຫາເຮັດໄດ້ງ່າຍ ເຊັ່ນ ຖ້າຈະເຕີມ ABOVE(ceiling, floor) ເຂົ້າໄປໃນຖານຄວາມຮູ້ ຈະໃສ່ປ່ອນໃດທີ່ຈະໄດ້ບໍ່ຕ້ອງບອກທຸກເທື່ອເມື່ອມີການກ່າວອ້າງເຖິງເລື່ອງຂອງຫ້ອງເພາະ ' ເພດານຢູ່ເໜືອຟື້ນ' ນີ້ເປັນຄວາມຈິງທີ່ໄປໃນເລື່ອງທີ່ກ່ຽວກັບຫ້ອງ.

ຂະບວນການດັ່ງກ່າວມາຂ້າງເທິງທັງໝົດ, ຄວາມຈິງແລ້ວຈະເປັນເລື່ອງທີ່ກ່ຽວກັບການສະແດງຄວາມຮູ້ໂດຍເຟມ(frame)ເຊິ່ງຈະໄດ້ມີການອະທິບາຍລະອຽດ ໃນບົດທີ 3

2.1.6 ຂະບວນການໃນການເລືອກກົດເກນ

ໂດຍປົກກະຕິແລ້ວວິທີໃນການເລືອກກົດເກນສາມາດເຮັດໄດ້ໂດຍວິທີການປຽບທຽບ(matching) ເຊິ່ງເຮັດໄດ້ໂດຍການນຳສະຖານະປັດຈຸບັນ(current state) ໄປປຽບທຽບກັບ ເງື່ອນໄຂຂອງກົດ ຂະບວນການຂອງການປຽບທຽບທີ່ນິຍົມໃຊ້ກັນຫຼາຍມີຢູ່ 2 ວິທີຄື:

1. ການເຮັດດັດສະນີ (Indexing)
2. ການຈັບຄູ່ກັບຕົວປ່ຽນ (Matching with variable)

2.1.6 ຂະບວນການໃນການເລືອກກົດເກນ

1. ການເຮັດດັດສະນີ (Indexing): ຈະເໝາະສົມສໍາລັບການສະແດງຄວາມຮູ້ໃນລະບົບການຜະລິດ ທີ່ວິທີການ ປ່ຽນສະຖານະຕ່າງ ໆ ໄດ້ຖືກກໍານົດໄວ້ເປັນເງື່ອນໄຂໄວ້ກ່ອນຕົວຢ່າງເຊັ່ນ ການແກ້ໄຂບັນຫາໂທນ້ຳ ເງື່ອນໄຂຕ່າງ ໆ ຖືກສ້າງໄວ້ກ່ອນແລະຖືກນຳມາໃຊ້ໃນຂັ້ນຕອນການຄົ້ນຫາຄໍາຕອບ ເພື່ອການອະທິບາຍວິທີການເຮັດດັດສະນີໃຫ້ຊັດເຈນຂຶ້ນ ຈຶ່ງຍົກຕົວຢ່າງດັ່ງນີ້ ໃນການແກ້ບັນຫາໂທນ້ຳນັ້ນ ສະຖານະເລີ່ມຕົ້ນຂອງນ້ຳໃນໂທທຽສອງເປັນ $(0, 0)$ ເພື່ອທີ່ຈະນຳໄປສູ່ສະຖານະ $(2, n)$ ໃນແຕ່ລະສະຖານະຈະມີກົດບາງຈຳນວນຖືກໃຊ້ເພື່ອປ່ຽນສະຖານະ ເຊັ່ນ ໃນສະຖານະເລີ່ມຕົ້ນ $(0, 0)$ ຈະມີກົດພຽງສອງກົດທີ່ນຳມາໃຊ້ໃນສະຖານະ ນີ້ໄດ້ ຄື ກົດທີ 1 ແລະ ກົດທີ 2

2.1.6 ຂະບວນການໃນການເລືອກກົດເກນ

2. ການຈັບຄູ່ຕົວປ່ຽນ (Matching variable)

ກຳນົດໃຫ້ປະໂຍກເຫລົ່ານີ້ເປັນຈິງ

1. ແດງໃຫຍ່ເປັນພໍ່ຂອງແດງ
2. ແດງເປັນພໍ່ຂອງແດງນ້ອຍ

ຫາກຕ້ອງການຮູ້ວ່າ ແດງໃຫຍ່ເປັນຫຍັງກັບແດງນ້ອຍ ຄອມພິວເຕີຍັງຕອບບໍ່ໄດ້ ແຕ່ຖ້າເຮົາໃຫ້ຄວາມຮູ້ ຫລື ກົດບາງຢ່າງທີ່ອະທິບາຍຄວາມສຳພັນລົງໄປ ເຊັ່ນ

ກົດ: ຖ້າ X ເປັນພໍ່ຂອງ Y ແລະ

Y ເປັນພໍ່ຂອງ Z ແລ້ວ X ເປັນປູ່ຂອງ Z

ດັ່ງນັ້ນໃນກະບວນການອະນຸມານ ຄວາມຮູ້ນີ້ ໃຫ້ X ເປັນແດງໃຫຍ່, Y ເປັນແດງ ແລະ Z ເປັນແດງນ້ອຍ ກໍຈະສະຫລຸບໄດ້ວ່າ ແດງໃຫຍ່ເປັນປູ່ຂອງ Z ກະບວນການນີ້ເຮົາຮຽກວ່າການປຽບທຽບຕົວປ່ຽນນັ້ນເອງ

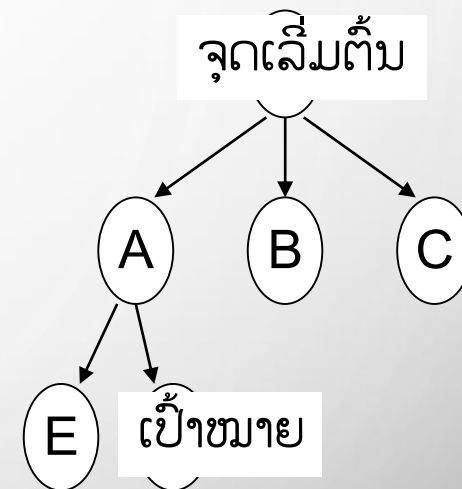
2.1.6 ຂະບວນການໃນການເລືອກກົດເກນ

ໃນການເລືອກກົດເກນ ນອກຈາກການພິຈາລະນາວ່າເຮົາຈະມີວິທີການເລືອກກົດເກນຂໍ້ໃດແລ້ວການເລືອກກົດເກນຍັງຈະຕ້ອງສົນໃຈການກຳນົດລຳດັບຂອງການພິຈາລະນາກົດໂດຍ ຄືການພິຈາລະນາວ່າເຮົາຈະພິຈາລະນາກົດຂໍ້ໃດກ່ອນ ຫຼື ຫຼັງເພາະລຳດັບຂອງການພິຈາລະນາກົດ ຈະມີຜົນຕໍ່ການແກ້ປັນຫາ, ໂດຍປົກກະຕິແລ້ວການກຳນົດລຳດັບຂອງການຄົ້ນຫາ ມີ 2 ຢ່າງຄື

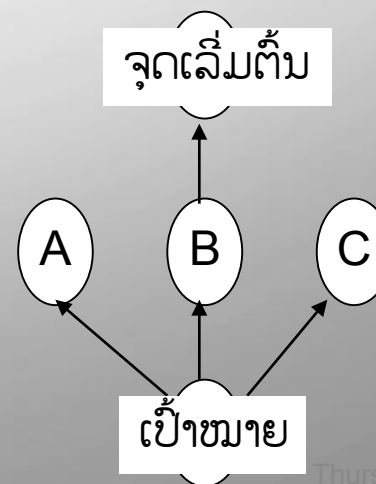
- ການຫາເຫດຜົນແບບໄປໜ້າ(Forward Reasoning) ແລະ
- ການຫາເຫດຜົນແບບຍ້ອນກັບຫຼັງ(Backward Reasoning)

2.1.6 ຂະບວນການໃນການເລືອກກົດເກນ

1. Forward reasoning ຈຸດເລີ່ມຕົ້ນ+ ການໃຊ້ກົດ-> ເປົ້າໝາຍ



2. Backward reasoning -- ເປົ້າໝາຍ+ ການໃຊ້ກົດ-> ຈຸດເລີ່ມຕົ້ນ

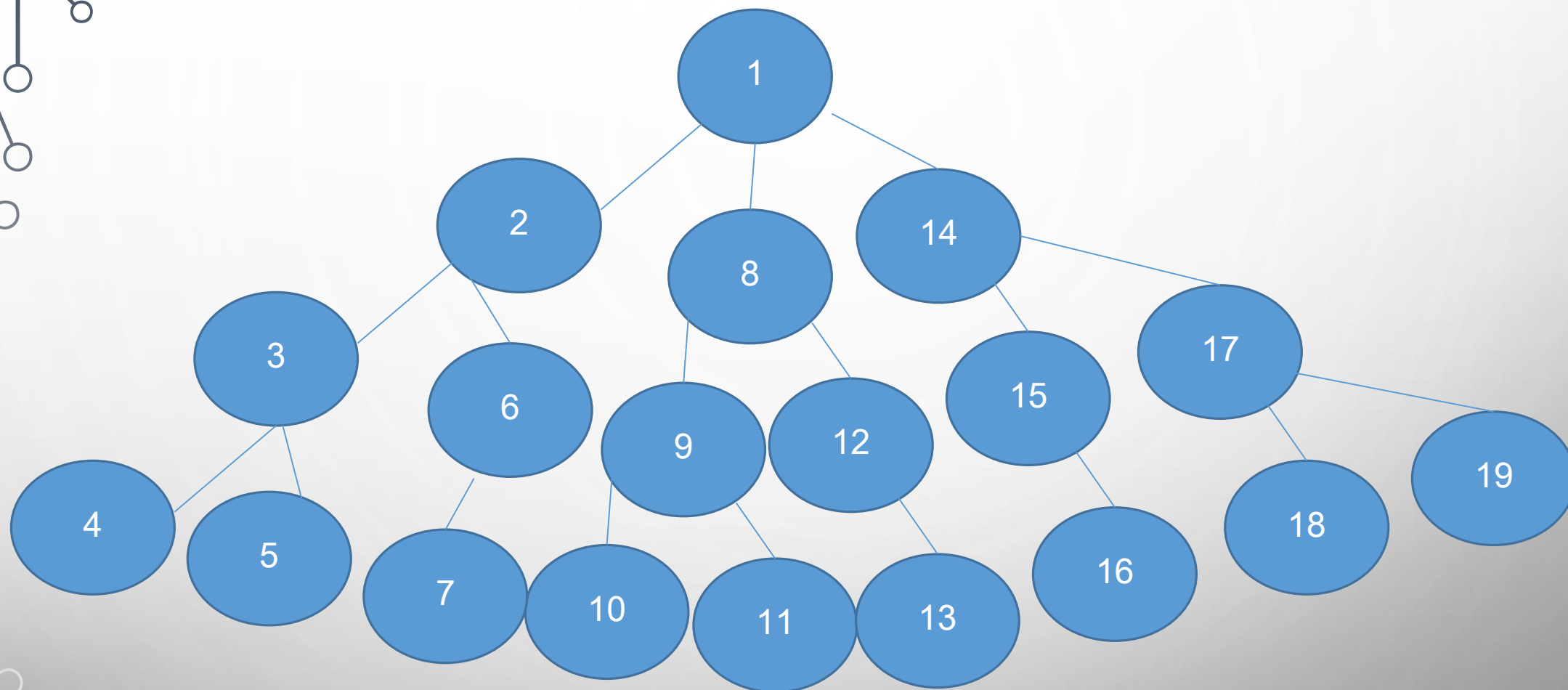


2.2. ການຄົ້ນຫາແບບງົມມືດ (BLIND SEARCH)

ການຄົ້ນຫາແບບງົມມືດ(Blind search) ເປັນການຄົ້ນຫາແບບທີ່ເດີນທາງຈາກໂນດໜຶ່ງໄປຍັງອີກໂນດໜຶ່ງ ໂດຍອາໄສທິດທາງເປັນຕົວກຳນົດການຄົ້ນຫາ ບໍ່ຕ້ອງມີຂໍ້ມູນຫຍັງມາຊ່ວຍໃນການຕັດສິນໃຈວ່າຈະເດີນທາງຕໍ່ໄປແບບໃດ ຫຼື ເວົ້າອີກແບບໜຶ່ງ ຄືການຈະເລືອກເອົາຂໍ້ມູນໃດມາຊ່ວຍໃນການຄົ້ນຫາຕໍ່ໄປບໍ່ຕ້ອງອາໄສຂໍ້ມູນໃດໆໝົດ ນອກຈາກທິດທາງ ຕົວຢ່າງຂອງການຄົ້ນຫາຂໍ້ມູນແບບນີ້ຄື **ການຄົ້ນຫາແບບເລິກກ່ອນ(Depth First Search)** ແລະ **ການຄົ້ນຫາແບບກວ້າງກ່ອນ (Breadth First Search)**

2.2.1 ການຄົ້ນຫາແບບລ່ວງເລິກກ່ອນ

ການຄົ້ນຫາແບບເລິກກ່ອນເປັນການຄົ້ນຫາທີ່ກຳນົດທິດທາງຈາກ ຮູບຂອງໂຄງສ້າງຕົ້ນໄມ້ ທີ່ເລີ່ມຕົ້ນຈາກໂນດຮາກ (Root node) ທີ່ຢູ່ເທິງສຸດແລ້ວລົງມາໃຫ້ເລິກທີ່ສຸດ ເມື່ອຮອດໂນດລຸ່ມສຸດ (Terminal Node) ໃຫ້ກັບຂຶ້ນມາທີ່ຈຸດສູງສຸດຂອງກິ່ງດ່ຽວກັນທີ່ ມີກິ່ງແຍກ ແລະ ຍັງບໍ່ໄດ້ແລ້ວເລີ່ມລົງໄປຈົນຮອດໂນດເລິກສຸດອີກ ເຮັດແບບນີ້ສະຫຼັບໄປເລື້ອຍໆຈົນພົບໂນດທີ່ຕ້ອງການຫາ ຫຼື ກວດ ສອບຄົບທຸກໂນດແລ້ວຕາມຮູບທີ່ 2.10 ການຄົ້ນຫາແບບເລິກກ່ອນ ຈະມີລຳດັບການໄປຕາມໂນດດັ່ງຕົວເລກທີ່ລະບຸໄວ້ໃນແຕ່ລະໂນດ



ຮູບທີ 2.10 ລຳດັບການຄົ້ນຫາແບບລວງເລິກກ່ອນແບບໂຄງສ້າງຕົ້ນໄມ້

2.2.1 ການຄົ້ນຫາແບບລ່ວງເລິກກ່ອນ

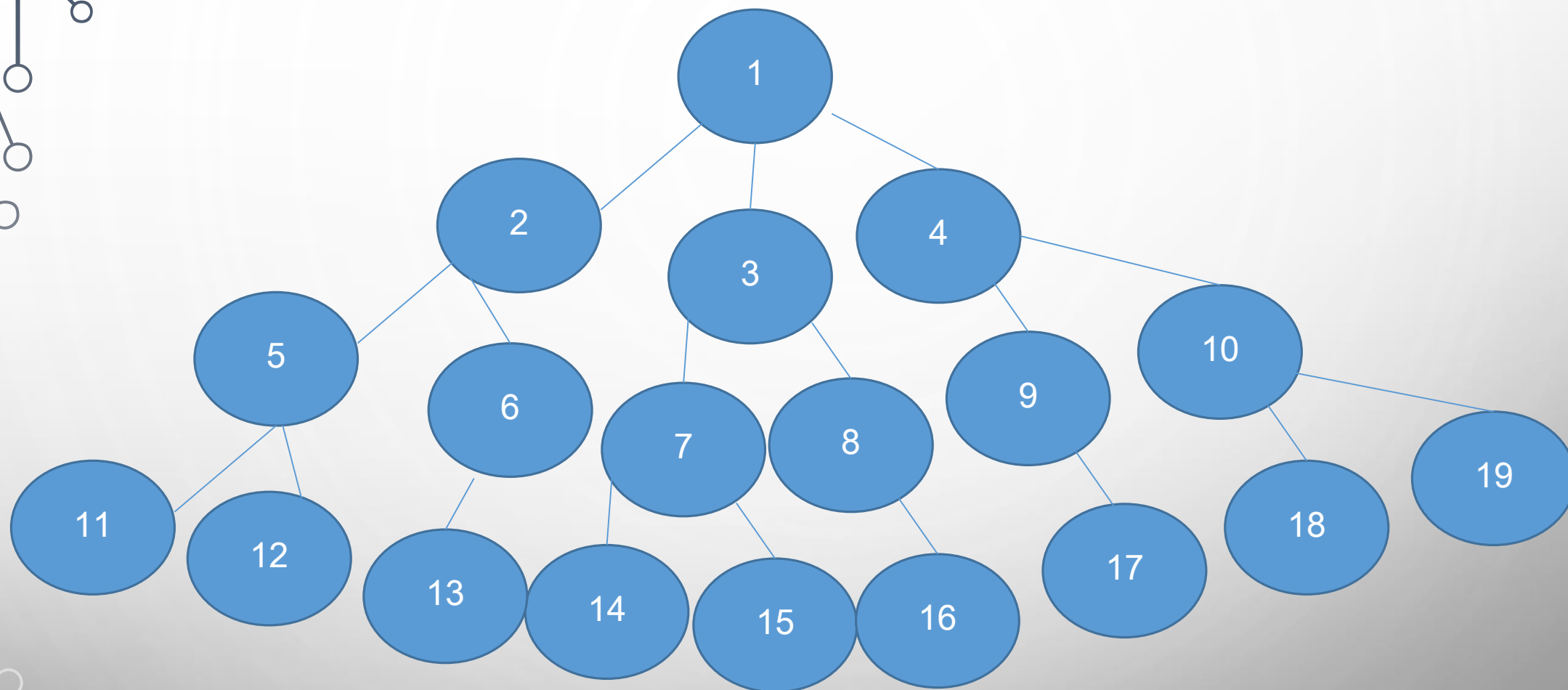
ອານາລິດທິມ ການຄົ້ນຫາຂໍ້ມູນແບບເລິກກ່ອນ

ໃຫ້ສະຖານະ 1 ໝາຍເຖິງໂນດທີ່ຍັງບໍ່ກວດສອບ, ສະຖານະ 2 ໝາຍເຖິງໂນດທີ່ຢູ່ໃນ STACK ແລະ ສະຖານະ 3 ໝາຍເຖິງໂນດທີ່ສ້າງການກວດສອບແລ້ວ

1. ສ້າງໃຫ້ໂນດທຸກໂນດມີສະຖານະເປັນ 1 ແລະນຳໂນດເລີ່ມຕົ້ນໄວ້ໃນ STACK ປ່ຽນສະຖານະເປັນ 2
2. ນຳໂນດເທິງສຸດໃນ STACK ອອກມາກວດສອບ ແລະປ່ຽນສະຖານະເປັນ 3
 - ຖ້າໂນດທີ່ສ້າງການກວດສອບຢູ່ຄືໂນດເປົ້າໝາຍລາຍງານໂນດທີ່ກວດສອບຄືໂນດເປົ້າໝາຍ ແລະໃຫ້ຂ້າມໄປທີ່ຂັ້ນຕອນ 4
 - ຖ້າໂນດທີ່ກວດສອບບໍ່ໃຊ້ໂນດເປົ້າໝາຍ ແລະຈຳນວນໂນດໃນ STACK ມີຫຼາຍກວ່າ 0 ໃຫ້ນຳໂນດຂ້າງຄຽງທີ່ມີສະຖານະເປັນ 1 ທັງໝົດ(ຖ້າມີ) ປ່ຽນສະຖານະເປັນ 2 ແລ້ວນຳໃສ່ໄວ້ໃນ STACK ແລະ ກັບໄປເຮັດຂໍ້ທີ່ 2
3. ລາຍງານການຄົ້ນຫາບໍ່ໄດ້
4. ການເຮັດວຽກສິ້ນສຸດ

2.2.2. ການຄົ້ນຫາແບບລວງກວ້າງກ່ອນ

ການຄົ້ນຫາແບບລວງກວ້າງກ່ອນເປັນການກຳນົດທິດທາງ
ການຄົ້ນຫາແບບເທື່ອລະລະດັບຂອງໂຄງສ້າງຕົ້ນໄມ້ໂດຍ
ເລີ່ມຈາກໂນດຮາກ (ລະດັບທີ 0) ແລ້ວລົງມາລະດັບທີ 1
ຈາກຊ້າຍໄປຂວາ ເມື່ອສຳເລັດລະດັບທີ 1 ໄປລະດັບທີ 2
ຈາກຊ້າຍໄປຂວາເຊັ່ນກັນ ເຮັດແບບນີ້ເລື້ອຍໆ ຈົນພົບ
ໂນດທີ່ຕ້ອງການຕາມຮູບທີ 2.12 ລຳດັບການເດີນທາງ
ຂອງໂນດເປັນໄປຕາມໝາຍເລກທີ່ລະບຸໄວ້ເທິງໂນດ



ຮູບທີ 2.12 ລຳດັບການຄົ້ນຫາແບບລວງກວ້າງກ່ອນແບບໂຄງສ້າງຕົ້ນໄມ້

2.2.2. ການຄົ້ນຫາແບບລວງກວ້າງກ່ອນ

ອານາຄົດທຶມ ການຄົ້ນຫາຂໍ້ມູນແບບລວງກວ້າງກ່ອນ

ໃຫ້ສະຖານະ 1 ໝາຍເຖິງໂນດທີ່ຍັງບໍ່ກວດສອບ, ສະຖານະ 2 ໝາຍເຖິງໂນດທີ່ຢູ່ໃນ QUEUE ແລະ ສະຖານະ 3 ໝາຍເຖິງໂນດທີ່ສ້າງການກວດສອບແລ້ວ

1. ທຳໃຫ້ໂນດທຸກໂນດມີສະຖານະເປັນ 1 ແລະນຳໂນດເລີ່ມຕົ້ນໄວ້ໃນ QUEUE ປ່ຽນສະຖານະເປັນ 2
2. ນຳໂນດທຳອິດໃນ QUEUE ອອກມາກວດສອບ ແລະປ່ຽນສະຖານະເປັນ 3
 - ຖ້າໂນດທີ່ສ້າງການກວດສອບຢູ່ຄືໂນດເປົ້າໝາຍ ລາຍງານໂນດທີ່ກວດສອບຄືໂນດເປົ້າໝາຍ ແລະ ໃຫ້ຂ້າມໄປທີ່ຂັ້ນຕອນ 4
 - ຖ້າໂນດທີ່ກວດສອບບໍ່ແມ່ນໂນດເປົ້າໝາຍ ແລະຈຳນວນໂນດໃນ QUEUE ມີຫຼາຍກວ່າ 0 ໃຫ້ນຳໂນດຂ້າງຄຽງທີ່ມີສະຖານະເປັນ 1 ທັງໝົດ(ຖ້າມີ) ປ່ຽນສະຖານະເປັນ 2 ແລ້ວນຳໃສ່ໄວ້ໃນ QUEUE ແລະກັບໄປຂໍ້ 2
3. ລາຍງານການຄົ້ນຫາທີ່ບໍ່ໄດ້
4. ການເຮັດວຽກສິ້ນສຸດ

2.3. ການຄົ້ນຫາແບບຮິວຣິສະຕິກ (HEURISTIC SEARCH)

ທາງດ້ານປັນຍາປະດິດ ການຄົ້ນຫາຄໍາຕອບອາໄສວິທີການທາງຮິວຣິສະຕິກ (Heuristic Search) ມີຄວາມແຕກຕ່າງຈາກການຄົ້ນຫາຂໍ້ມູນແບບທໍາມະດາຢູ່ທີ່ການຄົ້ນຫາຂໍ້ມູນແບບທໍາມະດາຜູ້ທີ່ສ້າງການຄົ້ນຫາຂໍ້ມູນຈະຕ້ອງກວດສອບຂໍ້ມູນເທື່ອລະຄັ້ງ, ທຸກຄັ້ງຈົນຄົບ ແຕ່ຮິວຣິສະຕິກຈະບໍ່ລົງໄປເບິ່ງຂໍ້ມູນທຸກຄັ້ງ, ວິທີການນີ້ຈະເລືອກໄດ້ຄໍາຕອບທີ່ເໝາະສົມໃຫ້ກັບການຄົ້ນຫາ ເຊິ່ງມີຂໍ້ດີຄື: ສາມາດສ້າງການຄົ້ນຫາຄໍາຕອບຈາກ ຂໍ້ມູນທີ່ມີຂະໜາດໃຫຍ່ຫຼາຍໆ ໄດ້ ແຕ່ຂໍ້ເສຍຄືຄໍາຕອບທີ່ໄດ້ເປັນພຽງຄໍາຕອບທີ່ດີເທົ່ານັ້ນ, ແຕ່ເນື່ອງຈາກວ່າປັນຫາໃນບາງລັກສະນະນັ້ນໃຫຍ່ຫຼາຍ ແລະ ເປັນໄປບໍ່ໄດ້ທີ່ຈະສ້າງການຄົ້ນຫາດ້ວຍວິທີແບບທໍາມະດາຂະບວນການຂອງຮິວຣິສະຕິກຈຶ່ງເປັນສິ່ງທີ່ຈໍາເປັນ.