Nonary Maze

André Antonitsch, Pedro Braga, Rovane Moura

[1]Faculdade de Informática (FACIN)
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

[andre.antonitsch, pedro.alves.002, rovane.moura@acad.pucrs.br

Resumo. *Nonary Maze é uma ideia de jogo do tipo puzzle proposta pelos autores, projetada e com protótipo desenvolvido. Neste documento serão apresentadas as definições formais utilizadas para comprovação de funcionalidade do laço principal do jogo, com as devidas propriedades e proposições utilizadas.*

1. Introdução

A base principal do jogo é composta por um labirinto formado por salas, onde cada sala possui certo número de portas numeradas com números entre 1 e 9. O labirinto possui uma sala inicial, onde começam 9 personagens, também numerados de 1 à 9, e o objetivo do jogo é fazer com que todos os personagens atinjam uma certa sala destino (ou sala objetivo) respeitando um conjunto de regras.

A primeira regra é que somente grupos de 3 à 5 personagens podem passar por uma porta de cada vez. A regra principal, que rege todo o objetivo deste documento e de sua formalização, é que dado este conjunto de personagens, para que eles passem por uma porta de número n, a raiz digital [Moura 2017][O'Beirne 1965] de seus números precisa ser igual a n. Esta propriedade está bem descrita na fórmula 26 na seção 3.

Nas seções seguintes, serão descritas formalmente as assinaturas de cada predicado e função, assim como as proposições fatos utilizados para a definição formal do labirinto e suas regras. Ao final temos a prova concreta sobre as regras de transição.

2. Restrições/Definições sobre o Modelo

2.1. Assinaturas

- $Player :: \mathbb{N}$
- $Door :: \mathbb{N} \times Room$
- $Room :: \{Door\} \times \{Door\}$
- $Maze :: \{Room\} \times Room \times Room \times (Player \to Room)$
- $Combo :: Maze \times \{Player\}$

2.2. Funções

- $pcode :: Player \to \mathbb{N}$
- $dcode :: Door \to \mathbb{N}$
- $destination :: Door \to Room$
- $doors, entrances :: Room \to \{Door\}$
- $rooms :: Maze \to \{Room\}$
- $start, goal :: Maze \to Room$

- $occupies :: Maze \rightarrow (Player \rightarrow Room)$
- $maze :: Combo \rightarrow Maze$
- $people :: Combo \rightarrow \{Player\}$
- $where :: Maze \times Player \rightarrow Room$
- $setSum :: \{\mathbb{N}\} \rightarrow \mathbb{N}$
- $ninesOut :: \mathbb{N} \rightarrow \mathbb{N}$
- $digitalRoot :: \{\mathbb{N}\} \rightarrow \mathbb{N}$
- $first, last :: Maze$
- $transition :: Maze \times Maze \times Room \times Player \rightarrow Bool$
- $admissionExam :: Maze \times Maze \rightarrow Bool$
- $next :: Maze \rightarrow Maze$

2.3. Proposições

$$\forall p : Player.(1 \leq pcode(p) \wedge pcode(p) \leq 9) \tag{1}$$

$$\forall d : Door.(1 \leq dcode(p) \wedge dcode(p) \leq 9) \tag{2}$$

$$\forall m : Maze.(start(m) \neq goal(m)) \tag{3}$$

$$\forall m : Maze.(|entrances(start(m))| = 0) \tag{4}$$

$$\forall m : Maze.(|doors(goal(m))| = 0) \tag{5}$$

$$\forall m : Maze.\forall r : Room.(r \in rooms(m)) \tag{6}$$

$$\forall m : Maze.\neg\exists r : (rooms(m) - goal).(|doors(r)| = 0) \tag{7}$$

$$\forall r : Room.(|doors(r)| \leq 3) \tag{8}$$

$$!\exists r : Room.(|doors(r)| = 1) \tag{9}$$

$$\begin{aligned} \forall m : Maze.\forall r : Room.\forall d : Door.((dcode(d) = 9 \rightarrow destination(d) = \\ goal(m)) \oplus (destination(d) \in rooms(m) - goal)) \end{aligned} \tag{10}$$

$$\forall c : Combo.(|people(c)| \geq 3 \wedge |people(c)| \leq 5) \tag{11}$$

$$\begin{aligned} \forall c : Combo.!\exists r : Room.\forall p : people(c).!\exists d : doors(r). \\ (where(maze(c), p) = r \wedge DigitalRoot(pcode(people(c))) = dcode(d)) \end{aligned} \tag{12}$$

2.4. Fatos

$$\forall p : Player.\forall f : occupies(first).(f(p) = start(first)) \tag{13}$$

$$\forall d : Door.!\exists r : Room.(d \in doors(r)) \tag{14}$$

$$\forall r : Room.\forall s : Room - r.(|doors(r) \cap doors(s)| = 0) \tag{15}$$

$$!\exists r : Room.(r \in destination(doors(r))) \tag{16}$$

$$\forall p1 : Player.\forall p2 : Player - p1.(pcode(p1) \neq pcode(p2)) \tag{17}$$

$$\forall d1 : Door.\forall d2 : Door - d1.(dcode(d1) \neq dcode(d2)) \tag{18}$$

$$\forall r : Room.(r \in destination(start(first))) \tag{19}$$

$$\forall m : Maze.(start(first) = start(m) \land goal(first) = goal(m)) \tag{20}$$

$$\forall r : Room.(DigitalRoot(dcode(doors(r))) = 9) \tag{21}$$

$$\forall r : Room.\forall d : Door.((r \in destination(d) \rightarrow d \in entrances(r)) \oplus \neg(d \in entrances(r))) \tag{22}$$

$$\forall r : Room - goal(last).!\exists r2 : Room.(destination(doors(r)) = r2) \tag{23}$$

$$\forall m : Maze.\forall f.occupies(m).\forall p : Player.((f(p) = goal(m) \rightarrow (\exists c : Combo.(maze(c) = m))) \oplus (\neg\exists c : Combo.(maze(c) = m)) \tag{24}$$

$$!\exists m : first.!\exists m' : next(m).(admissionExam(m, m')) \tag{25}$$

3. Prova - Validade do Labirinto

Para provar que é possível solucionar uma instância arbitrária do labirinto – onde solucionar significa levar todos os personagens à última sala do labirinto – é suficiente mostrar que é possível sair de uma sala com todos os personagens dada uma combinação arbitrária de portas que respeita à proposição 3, pois um labirinto é apenas uma sequência de salas sem bifurcações.

Considerando as proposições 1, 17, temos que o grupo de personagens possuem *player codes* de 1 a 9 sem repetições, de acordo com a definição de raiz digital, temos como premissa

$$DR(\sum pcodes_i) = 9 \tag{26}$$

E pela proposição 3, temos a premissa

$$DR(\sum dcodes_i) = 9 \tag{27}$$

onde $dcodes$ é o conjunto de *door codes* da sala escolhida arbitrariamente.

De acordo com a proposição 11, a quantidade de personagens que pode passar por uma porta varia de 3 a 5, e sobre a quantidade de portas, pode-se apenas ter 1 (única porta 9), 2 ou 3 portas por sala. Com base nisso, temos 3 casos a serem mostrados:

- 1 porta $\rightarrow$ 1 grupo de 4 e outro de 5 personagens.
- 2 portas $\rightarrow$ 1 grupo de 4 e outro de 5 personagens.
- 3 portas $\rightarrow$ 3 grupos de 3 personagens.

Para os casos com 1 e 2 portas, considerando dois conjuntos distintos $X = \{a, b, c, d\}$ e $Y = \{p, q, r, s, t\}$ de *player codes* cujas raízes digitais somam os valores necessários para se abrir as portas na sala escolhida arbitrariamente, é suficiente mostrar que

$$DR(\sum (X \cup Y)_i) = 9$$

O mesmo pode ser reescrito como

$$DR(\sum X_i + \sum Y_i) = 9$$

$$DR(DR(\sum X_i) + DR(\sum Y_i)) = 9$$

pela propriedade de decomposição da raiz digital. Logo apenas resta mostrar que

$$DR(DR(a + b + c + d) + DR(p + q + r + s + t)) = 9$$

$$DR(a + b + c + d + p + q + r + s + t) = 9$$

Por definição temos que $pcodes = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e que o conjunto $X \cup Y = \{a, b, c, d, p, q, r, s, t\}$ é um conjunto de *player codes* sem repetições, então podemos afirmar que $pcodes = X \cup Y = \{a, b, c, d, p, q, r, s, t\}$

$$DR(a + b + c + d + p + q + r + s + t) = DR(\sum pcodes_i)$$

E seguindo a premissa 26, prova-se que

$$DR(a + b + c + d + p + q + r + s + t) = 9$$

■

Para o caso de 3 portas, a prova é análoga ao caso de 1 e 2 portas, porém com a divisão do grupo de personagens em 3 subgrupos de 3 pessoas cada.

$$X = \{a, b, c\}, Y = \{d, p, q\}, Z = \{r, s, t\}$$

$$DR(\sum (X \cup Y \cup Z)_i) = 9$$

$$DR(\sum X_i + \sum Y_i + \sum Z_i) = 9$$

$$DR(DR(\sum X_i) + DR(\sum Y_i) + DR(\sum Z_i)) = 9$$

$$DR(DR(a + b + c) + DR(d + p + q) + DR(r + s + t)) = 9$$

$$DR(a + b + c + d + p + q + r + s + t) = 9$$

■

Referências

Moura, R. (2017). Raiz digital de elemento neutro 9.

O'Beirne, T. H. (1965). *Puzzles and paradoxes*. New York, Oxford University Press.