

Objectifs de Tests

1 *Stubs* utilisés

Les objets de test suivant peuvent-être référencés dans les cas de test :

1.1 Le terrain TER1

$\text{TER1} \stackrel{def}{=} \text{Terrain}::\text{init}(5, 3)$

Avec pour blocs :

.####
.OX.Y
....?

- X héro
- # mur
- . vide
- 0 rocher
- Y diamant
- ? sortie fermée

1.2 La position POS1

$\text{POS1} \stackrel{def}{=} \text{Position}::\text{init}(5, 3, 1, 1)$

2 Bloc

2.1 Couverture des préconditions

Aucune précondition dans Bloc.

2.2 Couverture des invariants

Objectif Bloc_invariant1 (minimisation de isVide)

Cas de test Bloc_invariant1 :
Préambule : B = init(VIDE, POS1)
Contenu : aucun
Oracle : isVide(B)

Objectif Bloc_invariant2 (minimisation de isSolide)

Cas de test Bloc_invariant2 :
Préambule : B = init(ROCHER, POS1)
Contenu : aucun
Oracle : isSolide(B)

Objectif Bloc_invariant3 (minimisation de isDeplacable)

Cas de test Bloc_invariant3 :
Préambule : B = init(ROCHER, POS1)
Contenu : aucun
Oracle : isDeplacable(B)

Objectif Bloc_invariant4 (minimisation de isTombable)

Cas de test Bloc_invariant4 :
Préambule : B = init(ROCHER, POS1)
Contenu : aucun
Oracle : isTombable(B)

Objectif Bloc_invariant5 (minimisation de isSortie)

Cas de test Bloc_invariant5 :
Préambule : B = init(SORTIE_OUVERTE, POS1)
Contenu : aucun
Oracle : isSortie(B)

Objectif Bloc_invariant6 (minimisation de isSortieFermee)

Cas de test Bloc_invariant6 :

Préambule : B = init(SORTIE_FERMEE, POS1)

Contenu : aucun

Oracle : isSortieFermee(B)

Objectif Bloc_invariant7 (minimisation de isHero)

Cas de test Bloc_invariant7 :

Préambule : B = init(HERO, POS1)

Contenu : aucun

Oracle : isHero(B)

Objectif Bloc_invariant8 (minimisation de isTerre)

Cas de test Bloc_invariant8 :

Préambule : B = init(TERRE, POS1)

Contenu : aucun

Oracle : isTerre(B)

2.3 Couverture des postconditions

Objectif Bloc_init_post1 (postcondition de init sur getType)

Cas de test Bloc_init_post1 :

Préambule : aucun

Contenu : B = init(HERO, POS1)

Oracle : getType(B) = HERO

Objectif Bloc_init_post2 (postcondition de init sur getPosition)

Cas de test Bloc_init_post2 :

Préambule : aucun

Contenu : B = init(HERO, POS1)

Oracle : getPosition(B) = POS1

Objectif Bloc_setType_post1 (postcondition de setType sur getType)

Cas de test Bloc_setType_post1 :

Préambule : B1 = init(HERO, POS1)

Contenu : B2 = setType(B1, ROCHER)

Oracle : getType(B2) = ROCHER

Objectif Bloc_setType_post2 (postcondition de setType sur getPosition)

Cas de test Bloc_setType_post2 :

Préambule : B1 = init(HERO, POS1)

Contenu : B2 = setType(B1, ROCHER)

Oracle : getPosition(B2) = POS1

2.4 Couverture des transitions

Objectif Bloc_setType_trans (transition de setType)

Cas de test Bloc_setType_trans :

Préambule : B1 = init(HERO, POS1)

Contenu : B2 = setType(B1, ROCHER)

Oracle : getType(B2) = ROCHER \wedge getPosition(B2) = POS1

3 Position

3.1 Couverture des préconditions

Objectif Position_init_pre (précondition de init)

Cas de test Position_init_pre_true :

Préambule : aucun

Contenu : aucun

Oracle : \exists MJ, MJ = init(5, 5, 2, 3)

Cas de test Position_init_pre_false1 :

Préambule : aucun

Contenu : aucun

Oracle : \nexists MJ, MJ = init(0, 5, 2, 3)

Cas de test Position_init_pre_false2 :

Préambule : aucun

Contenu : aucun

Oracle : \nexists MJ, MJ = init(5, 0, 2, 3)

Cas de test Position_init_pre_false3 :

Préambule : aucun

Contenu : aucun

Oracle : \nexists MJ, MJ = init(5, 5, -1, 3)

Cas de test Position_init_pre_false4 :

Préambule : aucun

Contenu : aucun

Oracle : \nexists MJ, MJ = init(5, 5, 2, -1)

3.2 Couverture des invariants

Aucun invariant dans Position.

3.3 Couverture des postconditions

Objectif Position_init_post1 (postcondition de init sur getLargeur)

Cas de test Position_init_post1 :

Préambule : aucun

Contenu : P = init(5, 4, 2, 3)

Oracle : getLargeur(P) = 5

Objectif Position_init_post2 (postcondition de init sur getHauteur)

Cas de test Position_init_post2 :

Préambule : aucun

Contenu : $P = \text{init}(5, 4, 2, 3)$

Oracle : $\text{getHauteur}(P) = 4$

Objectif Position_init_post3 (postcondition de init sur getX)

Cas de test Position_init_post3 :

Préambule : aucun

Contenu : $P = \text{init}(5, 4, 2, 3)$

Oracle : $\text{getX}(P) = 2 \% 5$

Objectif Position_init_post4 (postcondition de init sur getY)

Cas de test Position_init_post4 :

Préambule : aucun

Contenu : $P = \text{init}(5, 4, 2, 3)$

Oracle : $\text{getY}(P) = 3 \% 4$

Objectif Position_deplacerVersDirection_post1 (postcondition de deplacerVersDirection sur getX)

Cas de test Position_deplacerVersDirection_post1_1 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{GAUCHE})$

Oracle : $\text{getX}(P2) = (2 - 1) \% 5$

Cas de test Position_deplacerVersDirection_post1_2 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{DROITE})$

Oracle : $\text{getX}(P2) = (2 + 1) \% 5$

Cas de test Position_deplacerVersDirection_post1_3 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{HAUT})$

Oracle : $\text{getX}(P2) = 2$

Objectif Position_deplacerVersDirection_post2 (postcondition de deplacerVersDirection sur getY)

Cas de test Position_deplacerVersDirection_post2_1 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{HAUT})$

Oracle : $\text{getY}(P2) = (3 - 1) \% 4$

Cas de test Position_deplacerVersDirection_post2_2 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{BAS})$

Oracle : $\text{getY}(P2) = (3 + 1) \% 4$

Cas de test Position_deplacerVersDirection_post2_3 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{GAUCHE})$

Oracle : $\text{getY}(P2) = 3$

3.4 Couverture des transitions

Objectif Position_deplacerVersDirection_trans (transition de deplacerVersDirection)

Cas de test Position_deplacerVersDirection_trans1 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{HAUT})$

Oracle : $\text{getY}(P2) = (3 - 1) \% 4 \wedge \text{getX}(P2) = 2$

Cas de test Position_deplacerVersDirection_trans2 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{BAS})$

Oracle : $\text{getY}(P2) = (3 + 1) \% 4 \wedge \text{getX}(P2) = 2$

Cas de test Position_deplacerVersDirection_trans3 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{GAUCHE})$

Oracle : $\text{getY}(P2) = 3 \wedge \text{getX}(P2) = (2 - 1) \% 5$

Cas de test Position_deplacerVersDirection_trans4 :

Préambule : $P1 = \text{init}(5, 4, 2, 3)$

Contenu : $P2 = \text{deplacerVersDirection}(P1, \text{DROITE})$

Oracle : $\text{getY}(P2) = 3 \wedge \text{getX}(P2) = (2 + 1) \% 5$