

# Project MCS: Logical Pacman

## Iteratie 1

Jessa Bekker  
???

Karel Moens  
s0215430

10 november 2013

## 1 Inleiding

In dit verslag worden kort de design keuzes bij het uitwerken van het eerste deel van het project toegelicht. De opdracht was een moment opname van een pacman spel in IDP te modelleren. Zowel een algemene aanpak als een aantal opmerkelijke predicaten worden besproken. Tot slot vermeldt de tekst ook de tijd die nodig was om dit te verwezenlijken.

## 2 Implementatiestrategie

Bij het begin van de uitwerking werd eerst gekeken naar wat gegeven werd en wat er daarnaast nog verder gedefinieerd zou moeten worden. Vervolgens werden alle vereisten en restrictie op het systeem een voor een in eerste orde logica en inductieve definities vertaald.

**Transition** Om niet telkens voor elke richting een verschillend geval te moeten maken, werd een *Transition* predicaat gedefinieerd. Dit predicaat voegt de kennis over het verband tussen richtingen en de overgang van vakjes en coördinaten aan het systeem toe. Zo wordt het bijvoorbeeld duidelijk dat bij een vakje boven een ander, de x-coördinaat 1 groter is.

**Opposite Direction** Het vervatten van het begrip van tegengestelde richtingen laat toe om nog verder te abstraheren en predicaten compacter te beschrijven. Omdat elke richting een en slechts een tegengestelde heeft, leek de keuze voor een functie vanzelfsprekend. *OppositeDir* had via *Transition*

beschreven kunnen worden maar er werd gekozen voor een iets langere maar meer intuïtieve en leesbare implementatie.

**Position** Hoewel *NoPos* voldoende was om het volledige systeem te beschrijven, zijn dubbele negaties niet de meest leesbare constructie. Daarom werd geopteerd om ook een predicaat *Pos* als tegengestelde van *NoPos* in te voeren.

**IsWall** *IsWall* vormt de symmetrische uitbreiding van het gegeven *Wall* predicaat. Dit werd opnieuw geschreven om de code beter begrijpelijk te maken. Zowel hier als bij *Pos* werden definities gebruikt om te voorkomen dat het systeem nieuwe muren of posities creëert waar deze niet bedoeld zijn.

**Reach** Om op te dringen dat alle vakjes toegankelijk zijn vanaf elk ander vakje, werd inductie gebruikt. Ten eerste is een vakje bereikbaar vanaf dat zelfde vakje en alle vakjes er rond waartussen geen muren staan. Door transitiviteit is er ook een pad tussen twee vakjes, wanneer er een derde vakje bestaat dat verbonden is met zowel het begin- als het eindpunt van het pad.

### 3 Tijdsbesteding

### 4 Besluit

Een toestand in een spel pacman werd gemodelleerd door elke restrictie en algemene waarheid in eerste orde logica en inductieve definities te vertalen. Hiervoor werden een aantal nieuwe predicaten en functies gedefinieerd.