

Triskilian

Ted Schmidt
Auke-Dirk Pietersma

April 3, 2008



Inleiding

Als eind project voor het vak MAS wilde we graag een variant op het bekende zeeslag maken. Hierbij zullen drie agenten tegen elkaar strijden met als doel om uiteindelijk als laatste over te blijven. De keuze om 3 agenten in het systeem op te nemen zorgt er voor dat er meer kennis/strategie nodig is dan bij het standaard zeeslag.





Spelregels

In het spel Triskilian dienen 3 spelers te strijden voor de overwinning. Dit doet men door te proberen zo snelmogelijk alle schepen van de tegenstander te doen zinken. Wanneer het spel start zal iedere speler zijn schepen verdelen over het speelbord. Dit gebeurt door schepen op bepaalde coördinaten te plaatsen. Een schip van 4 lang zou bijvoorbeeld op de coördinaten $\{(1,1)(1,2)(1,3)(1,4)\}$ kunnen liggen. Wanneer deze geplaatst zijn kan de strijd beginnen. De spelers mogen om beurten bepalen op welke tegenstanders ze schieten en de locatie waar ze denken een schip te raken. Omdat alle schepen op een $N \times N$ bord verdeeld zijn, zal een schot 2 coördinaten hebben (x, y) . Wanneer alle schepen van speler zijn gezonken zal deze speler niet meer aan het spel meedoen. De speler die als laatste nog een of meerdere schepen overhoudt heeft gewonnen.

Triskilian

In Triskilian zullen de agenten om beurten op een tegenstander schieten. De tegenstanders op wie geschoten wordt kan willekeurig gekozen worden. Uiteraard wil elke agent winnen, zodat het belangrijk wordt een goede strategie te hanteren. Als eerste is de state waarin ons systeem verkeerd erg belangrijk. De state binnen ons systeem correspondeert met de huidige windrichting $State = \{Tris_1, Tris_2, Tris_3\}$. Deze windrichting is Common-knowledge voor ieder agent in het systeem. De windrichting is bepalend voor hoeveel kennis je aan je tegenstanders bloot legt. Wanneer er met een kanon geschoten wordt, en de windrichting waait in de richting van je tegenstanders dan zal er enige informatie bekend worden over waar het schot vandaan kwam. Dit leidt tot een kennis overdracht die er als volgt uit ziet.

$Agent_{schutter}$ weet de windrichting, dus hij weet ook welke agent zijn schot kan detecteren, zodat hij weet dat $Agent_x$ ook weet waar een van zijn schepen ligt (je schiet namelijk vanuit een schip).

Elke agent heeft een dataset van wat hij weet of denkt te weten over zijn tegenstanders, en wat hij weet/ denkt te weten wat tegenstanders over hem weten.

Het is bij Triskilian erg belangrijk om goed te overwegen op wie en waarom schiet ik!



Waarom Epistemische logica?

Epistemische logica is de logica van de 'kennis'. Kennis is bij de meeste strategische spellen het grootste goed, zo ook bij Triskilian. Als een agent als winnaar uit de strijd wil komen zal hij goed gebruik moeten maken van wat hij weet van de andere agenten en wat zij van hem weten. Voor het definiëren/modelleren van deze kennisstromen en mogelijke situaties/gevolgen is epistemische logica heel goed toepasbaar.

We zitten in een situatie waarin het agent A zijn beurt is om te schieten. Hij zal zijn redeneren als volgt beginnen. Zijn er al andere agenten die een schip van mij ontdekt hebben (tactisch is het handig om hiermee te schieten).

$\psi = \exists K_i \text{ locatie(schip)}$

Wanneer ψ onwaar blijkt te zijn kan de agent een willekeurig schip kiezen om mee te schieten. Vervolgens moet de windrichting overwogen worden, je weet namelijk wie welke informatie van jou gaat ontvangen. Wanneer agent A weet dat B weet waar een schip van A ligt dan is het een goede keuze om dat schip te nemen, omdat de totale kennis van agent B over agent A niet veel groter wordt. De kennis van B kan wel groter worden, maar dit ligt op een ander vlak. In een gunstige windrichting voor agent A, weet agent B dat agent A weet of hij raak of niet raakt geschoten heeft.

De hierboven geschetste weergave geeft al heel goed weer dat kennis logica heel goed toepasbaar is voor het beredeneren binnen ons systeem.

Agenten

Er zijn drie verschillende agenten, elke agent zal zijn eigen strategie geïmplementeerd krijgen. Verder is het voor alle agenten gelijk dat ze kennis verkrijgen via het systeem afhankelijk van de state van het systeem.

Implementatie

Dit zal waarschijnlijk een client-server model worden in java. Dit biedt ook mogelijkheden to het implementeren van verschillende interfaces (C#, applets, terminal, etc).



Model

Voor het modelleren van de kennis van agents wordt er gebruik gemaakt van epistemische logica. De epistemische taal wordt uitgebreid met:

- A , een set van m agenten $A = 1, 2, 4$
- b_p^m , een atoom die de boot positie van agent m representeert
- w_m , een atoom die de windrichting naar een agent m representeert



Kennis

Wanneer het spel begint heeft elke agent agent alleen kennis over zijn eigen bootposities en als common-knowledge de windrichting. Windrichting 1,2 betekend dat agent 1 en 2 informatie krijgen.

Tijdens het spel houdt elke agent kennis bij over:

- De bootposities van alle agents
- Wat alle agents weten over bootposities van alle andere agent
- Wat alle agents weten over wat andere agents weten over bootposities van andere agent.

Een voorbeeld van het verkrijgen van kennis tijdens het spel wordt hieronder voor agent 1 weergegeven.

Start van het spel:

Elke agent heeft kennis over de windrichting $\models C(W_{1,2} \vee W_{1,3} \vee W_{2,3})$

Agent 1 heeft alleen kennis over zijn eigen bootposities:

$(M,s) \models K_1 b^1_p$ waar $p = 1, \dots, \text{totaal aantal bootposities}$

Agent 1 schiet naar agent 2 op positie p:

Er zijn 3 mogelijke windrichtingen per windrichting 2 uitkomsten, raak of mis:

$W_{1,2}$:

- | | |
|---------------------|--|
| a) $K_1(\neg)b^2_p$ | agent 1 weet dat er op positie p (g)een boot staat |
| b) $K_1K_2b^1_p$ | agent 1 weet dat agent 2 zijn positie p weet |
| c) $K_1K_2K_1b^2_p$ | agent 1 weet dat agent 2 weet dat a) |

$W_{1,3}$:

- | | |
|---------------------|--|
| a) $K_1(\neg)b^2_p$ | agent 1 weet dat er op positie p (g)een boot staat |
| b) $K_1K_3b^1_p$ | agent 1 weet dat agent 3 zijn positie p weet |
| b) $K_1K_3b^2_p$ | agent 1 weet dat agent 3 agent 2 zijn positie p weet |
| c) $K_1K_3K_1b^2_p$ | agent 1 weet dat agent 3 weet dat a) |

$W_{2,3}$:

- | | |
|--------------------------|---|
| a) $\neg K_1(\neg)b^2_p$ | agent 1 weet <u>niet</u> of er op positie p (g)een boot staat |
| b) $K_1K_2b^1_p$ | agent 1 weet dat agent 2 zijn positie p weet |
| b) $K_1K_3b^1_p$ | agent 1 weet dat agent 3 zijn positie p weet |
| b) $K_1K_3b^2_p$ | agent 1 weet dat agent 3 agent 2 zijn positie p weet |
| c) $K_1K_2K_3b^1_p$ | agent 1 weet dat agent 2 weet dat agent 3 agent 1 zijn positie p weet |
| c) $K_1K_2K_3b^2_p$ | agent 1 weet dat agent 2 weet dat agent 3 agent 2 zijn positie p weet |
| c) $K_1K_3K_2b^1_p$ | agent 1 weet dat agent 3 weet dat agent 2 agent 1 zijn positie p weet |

Doordat alle agents positie p van agent 1 weten is positie p common-knowledge geworden.