

# Probabilistic programming for game content generation

Probleemstelling

---

*Robin Haveneers  
o.l.v. Angelika Kimmig & Luc De Raedt*

# Inhoudstafel

2

**1** Observatie

**2** Vraagstelling

**3** Waarom

**4** Hypothese

**5** Evidence

**6** Vereisten

## › Observatie

- © PCG volgens een declaratief paradigma bestaat reeds in de vorm van answer set programming (ASP)
- © Deze ASP-techniek wordt dan voorgesteld in AnsProlog, een variant op ProbLog
- © Deze methodes leiden tot goede uitkomsten.

## › Vraagstelling

© Kan ProbLog hier ook op worden toegepast, meer bepaald:

- Kan de “randomness” van deze methodes in ASP via AnsProlog vervangen worden door kansverdelingen in ProbLog?
- Biedt ProbLog en zijn kans-implementatie (en ook bv. het niet uniform zijn van kansen) een voordeel t.o.v. de “kansen” in AnsProlog?
- Kan het huidige ProbLog-systeem dit aan?

## › Waarom

- © Zoals eerder : ProbLog kan ook niet-uniforme kansverdelingen simuleren wat AnsProlog niet kan.
- © Deze “echte” kansverdeling in ProbLog kan er voor zorgen dat bepaalde bijna symmetrische/gelijke situaties kunnen worden vermeden.
- © Interessant toepassingsgebied van ProbLog.

## › Hypothese

© ProbLog zal hier inderdaad voor kunnen worden gebruikt.

→ ProbLog kan ervoor zorgen dat bepaalde feiten bijvoorbeeld normaal verdeeld zijn (ipv. uniform), waarmee bepaalde situaties beter kunnen worden voorgesteld.

## › Evidence

- © AnsProlog kan aan PCG doen en de conversie tussen AnsProlog en ProbLog is mogelijk.
- © ProbLog biedt dus alle mogelijkheden die (Ans)Prolog biedt en zou dus geen hinder mogen zijn.
- © Ik heb de “chromatic maze” code omgezet in ProbLog.

## › Vereisten

- © Degelijke kennis van ProLog en (nadien) ook ProbLog, en in het algemeen LP. (Kan in het begin verwarrend zijn)
- © Eenvoudig grafisch kunnen voorstellen van gegeneerde werelden. Kan niet ('native') in ProbLog zoals andere programmeertalen.
- © AnsProlog vlot kunnen omzetten in ProbLog



Einde