KU LEUVEN

Generatie van spelinhoud door middel van probabilistisch programmeren

Een analyse van de toepasbaarheid van ProbLog

Robin Haveneers - Begeleider: Dr. Angelika Kimmig

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

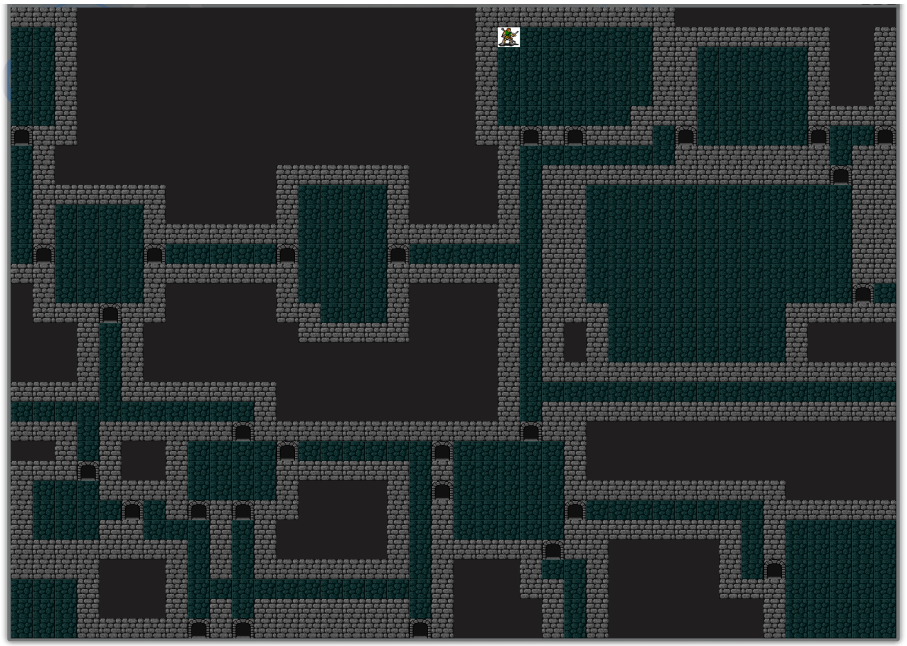
Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Overzicht

Generatie van spelinhoud

- Engels: Game Content Generation
- Rekwisieten (altaars, wapens, muntjes …)
- Werelden (platforms, kerkers …)
- ≠ Procedural Game Content Generation



KU LEUVEN

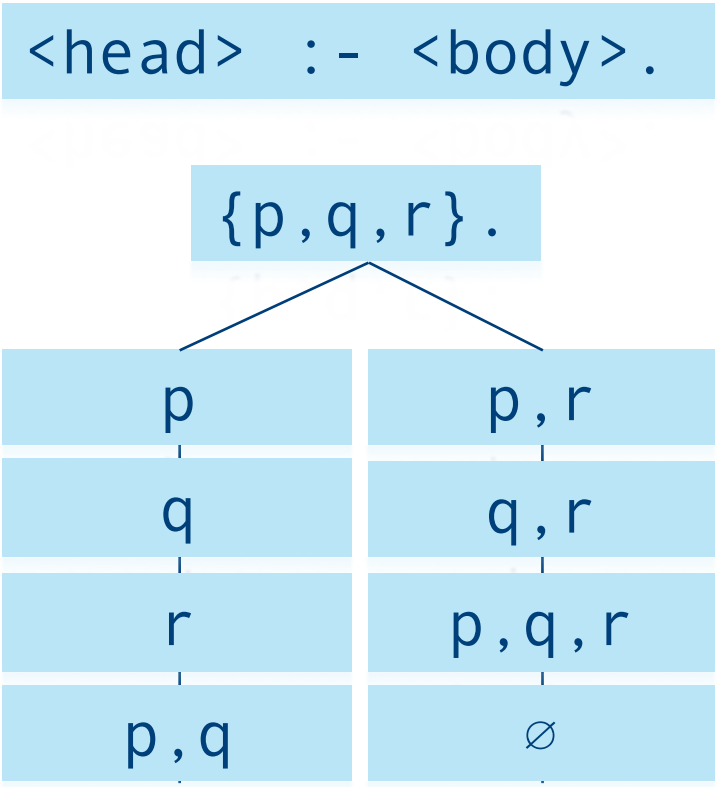
Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Overzicht

AnsProlog

- Answer Set Programming
  - Declaratief programmeren
  - Moeilijke (NP-harde) zoekproblemen
- AnsProlog = Syntax
- Solvers zoals Lparse, Clasp, DLV …



Overzicht

AnsProlog

```
{p,q,r}.
```
=
$$(p \lor \neg p) \land (q \lor \neg q) \land (r \lor \neg r)$$

```
1{p,q,r}2.
```
=
$$(p \lor \neg p) \land (q \lor \neg q) \land (r \lor \neg r) \land (p \lor q \lor r) \land \neg(p \lor q \lor r)$$

```
a(1..3).
```
=
```
a(1).
a(2).
a(3).
```

Overzicht

AnsProlog

- Clasp
  - Answer Set Solver
  - Los van oudere SAT- of ASP-solvers
  - Modelleringscapaciteit ASP
  - State-of-the-art boolean solver
- Clingo
  - Grounding (“variabel-vrij” maken)
  - Clasp
- Potassco: Potsdam Answer Set Solving Collection (University of Potsdam, Duitsland)

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Overzicht

ProbLog

- Probabilistic Logic Programming
  - Ontwikkeld hier door DTAI
  - Uitbreiding op ProLog
- Clauses annoteren met kansen
- Evidence
- ‘Learning’ uit partiële interpretaties
- ProbLog Versie 2 (d-DNNF ipv. BDD)

DTAI

Declaratieve Talen en Artificiële Intelligentie

```
p::f(X1,X2,....,Xn)  :- body
```

```
select_uniform(...)
```

```
select_weighted(...)
```

```
evidence(...)
```

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Probleemstelling

Hoofdstelling

- ProbLog
  - Kracht van het huidige ProbLog systeem

Verder ook

- ASP-constraints voorstellen in ProbLog
- Voordelen met (niet)-uniforme kansen
  - → Biedt randomness extra mogelijkheden

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Motivatie

- Nieuwe mogelijkheden ontdekken bij gebruik ProbLog
- Analyse ten opzichte van bestaande systemen
- Beperkingen ProbLog opsporen
- Uitbreiding onderzoeksdomein ProbLog

Overzicht

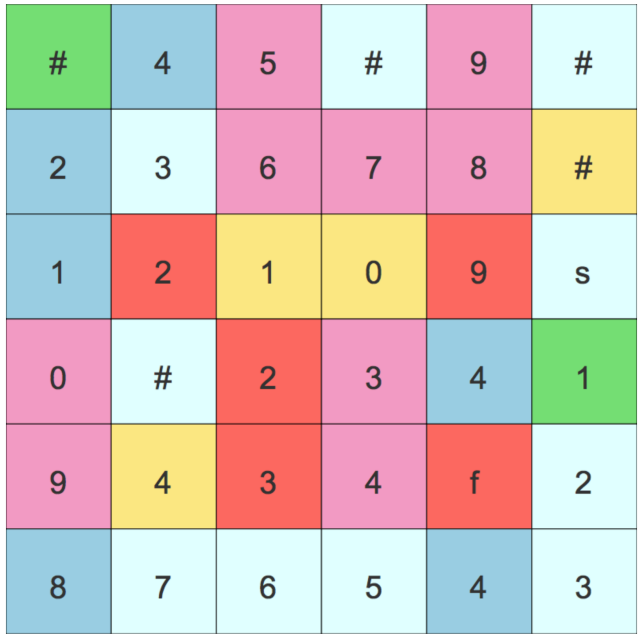
- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Aanpak

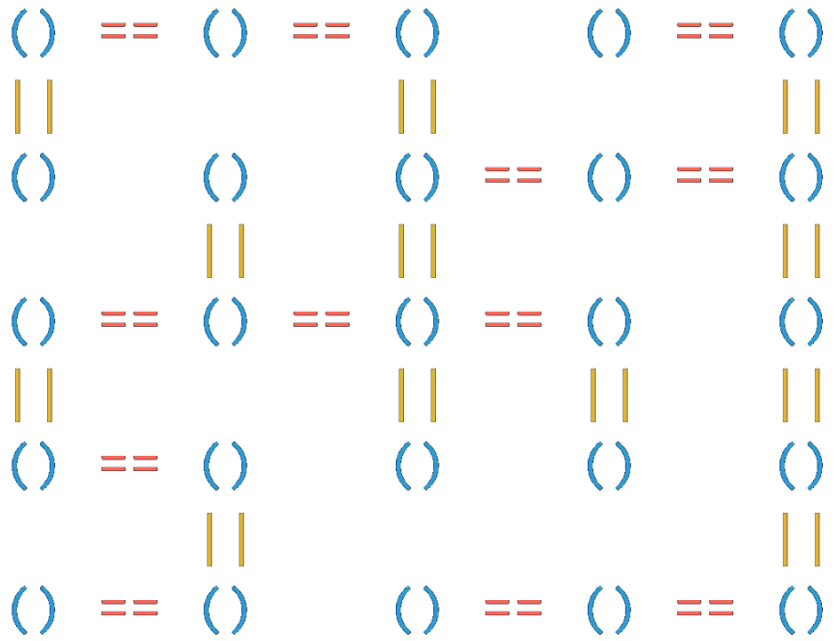
Algemeen

- Aan de hand van diverse soorten puzzels met elk een eigen visualiseertool

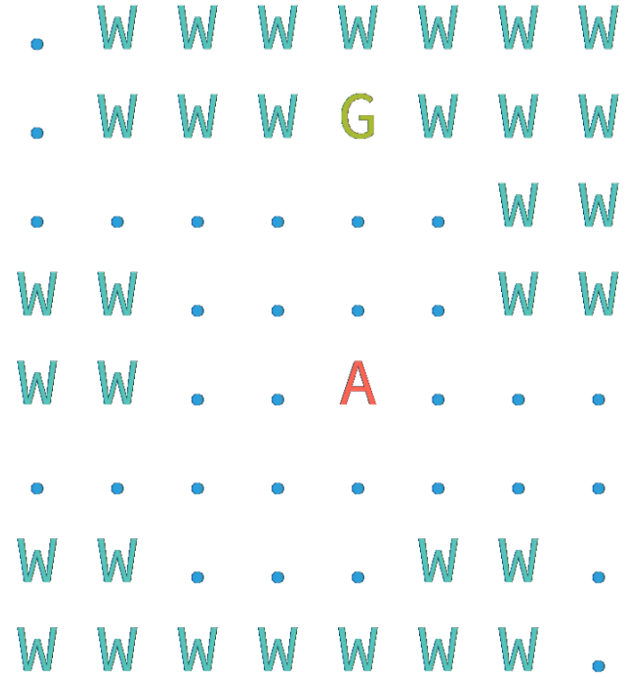
Chromatic Maze



Perfect Maze



Dungeon



KU LEUVEN

Aanpak

Code omzetten

Relatief eenvoudige constraints

- `dim(1..B) :- size(B)` →

  ```
  dim(D) :- size(B), between(1,B,D).
  ```

- `1{p,q,r}1` = Selecteer precies 1 uit `{p,q,r}` →

  ```
  select_uniform(I,[p,q,r],Result, Rest).
  ```

Aanpak

Code omzetten

lets complexere constraints

- `1{p,q,r}2` = Selecteer 1 of 2 uit `{p,q,r}` →

  `select_uniform(I,L,Result, Rest)).`

  `[p,q,r,(p,q),(p,r),(q,r),(p,q,r)]`

- `:- dit_kan_nooit.` → afhankelijk van context !

Aanpak

Code omzetten

Evidence-based constraints

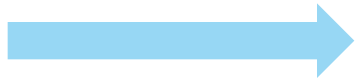
- Voorbeeld: chromatic maze

```
victory_at(T) :- player_at(T,X,Y), finish(X,Y).

victory :- victory_at(T).

:- victory_at(T), min_sol(MinSol),T < MinSol.
:- victory_at(T), max_sol(MaxSol),MaxSol < T.
:- not victory.
```

"Victory treedt op als de player op een bepaald moment T op het finish-vakje komt"



"Victory moet optreden en moet bereikt worden in minder dan MaxSol en meer dan MinSol tijd"

```
victory_at(T) :- player_at(T,X,Y), finish(X,Y).

victory :- victory_at(T).

:- victory_at(T), min_sol(MinSol),T < MinSol.
:- victory_at(T), max_sol(MaxSol),MaxSol < T.
:- not victory.
```

```
victory :- victory_at(T), time(T).

victory_at(T) :-
        time(T),
        min_sol(MinSol),
        max_sol(MaxSol),
        finish(X,Y),
        T > MinSol,
        T < MaxSol
        player_at(T, X, Y).

evidence(victory).
```

Aanpak

Code omzetten

Evidence-based constraints

- Voorbeeld: dungeons

"Elke tile is ofwel een wall, ofwel een gem ofwel een altar"

```
0 { sprite(T,wall);gem(T);altar(T) } 1 :- tile(T).

:- not 1 { altar(T) } 1.
:- not 1 { gem(T) } 1
```

"Er is ten hoogste 1 gem alsook 1 altar"

KU LEUVEN

```
0 { sprite(T,wall);gem(T);altar(T) } 1 :-
tile(T).

:- not 1 { altar(T) } 1.
:- not 1 { gem(T) } 1
```

```
only_one :-
findall((X,Y),(tile((X,Y)),sprite((X,Y),(X,Y,gem))), R),
length(R,1),
findall((Z,Q),(tile((Z,Q)),sprite((Z,Q),(Z,Q,altar))), T),
length(T,1).

evidence(only_one).
```

"Zoek alle gems, dit mag er maximaal 1 zijn. Zoek alle altaars, dit mag er maximaal 1 zijn."

Welke methode is nu beter: evidence of ‘alternatieven’ zoeken?

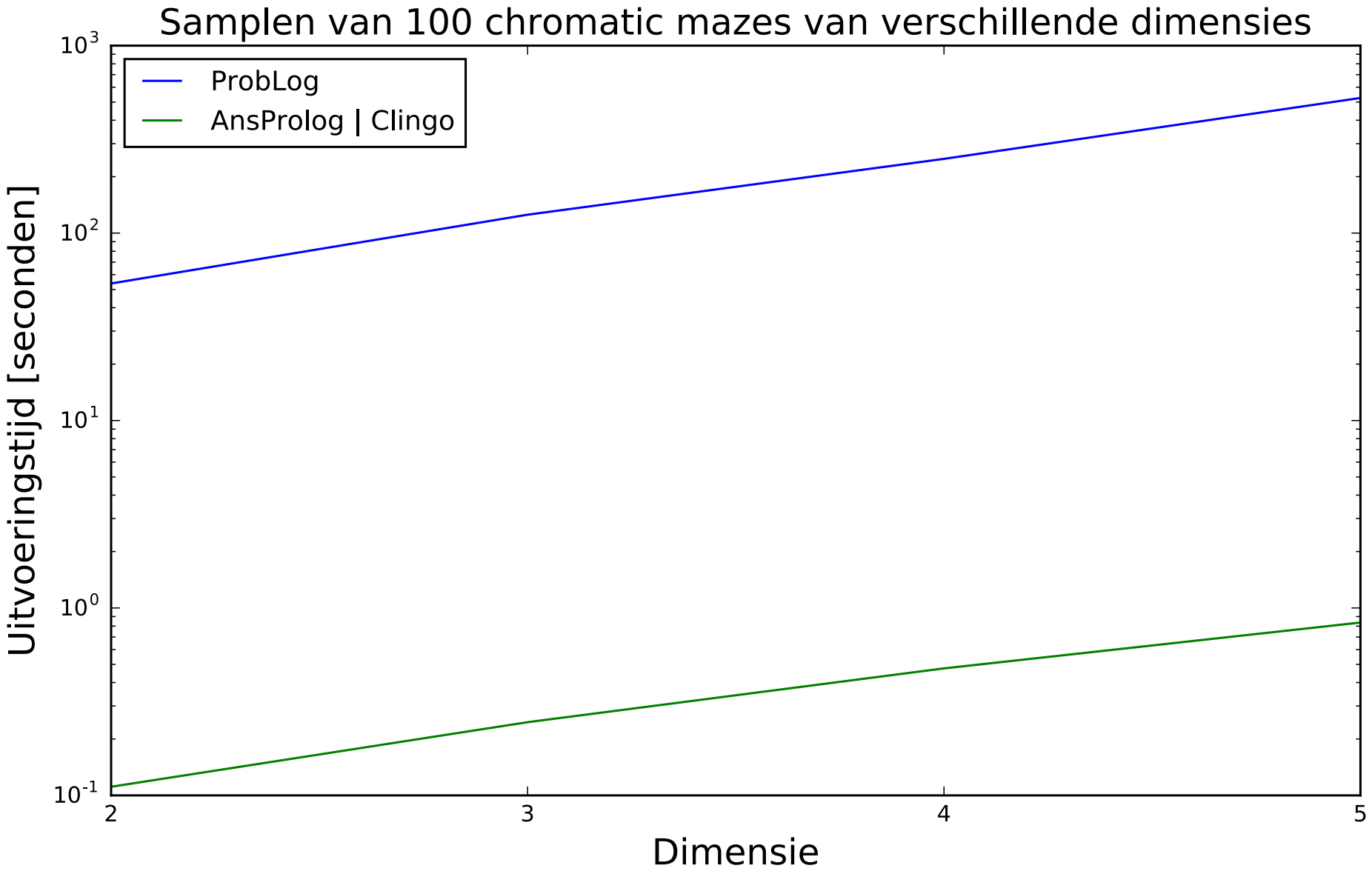
Overzicht

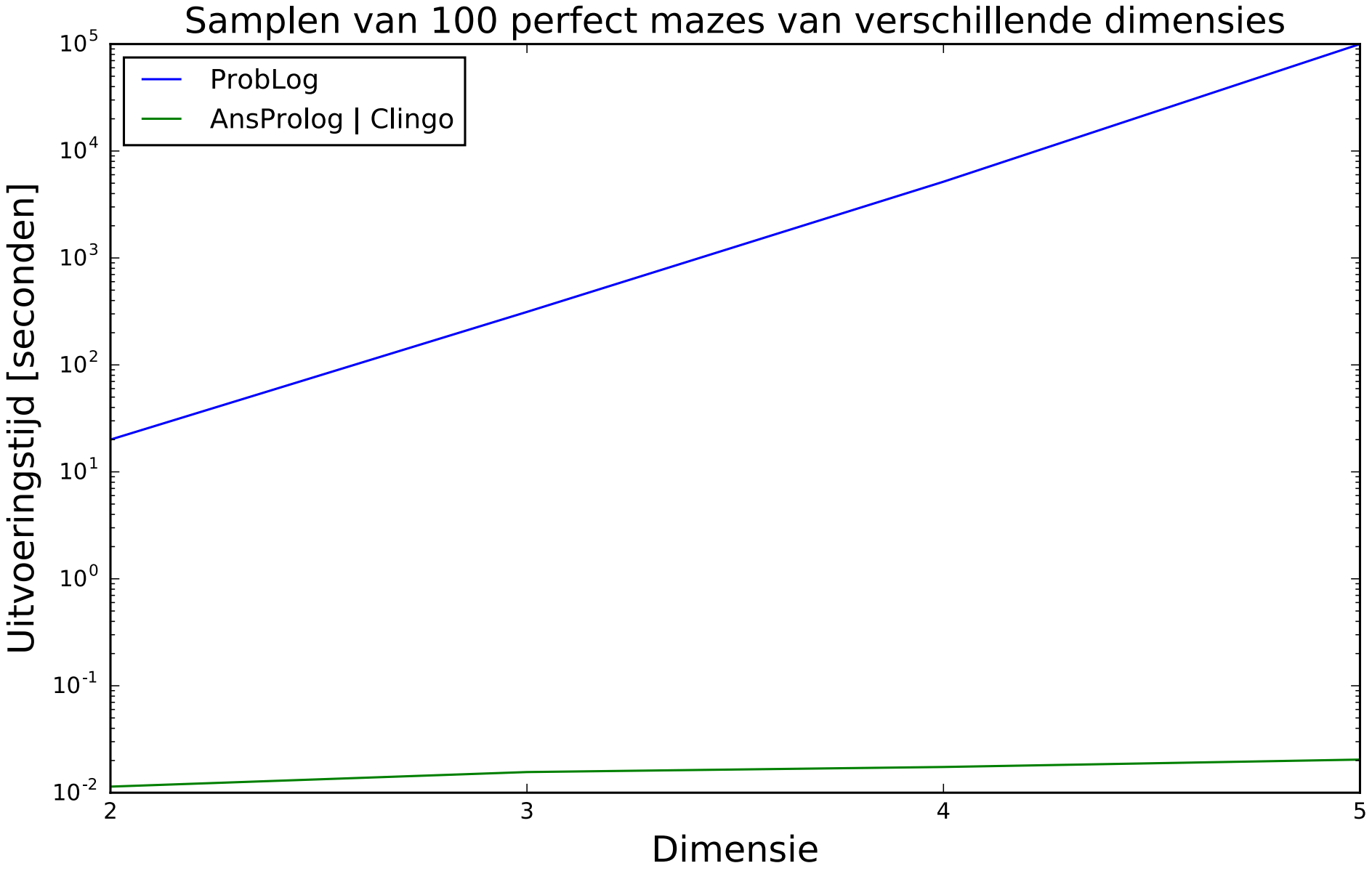
- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

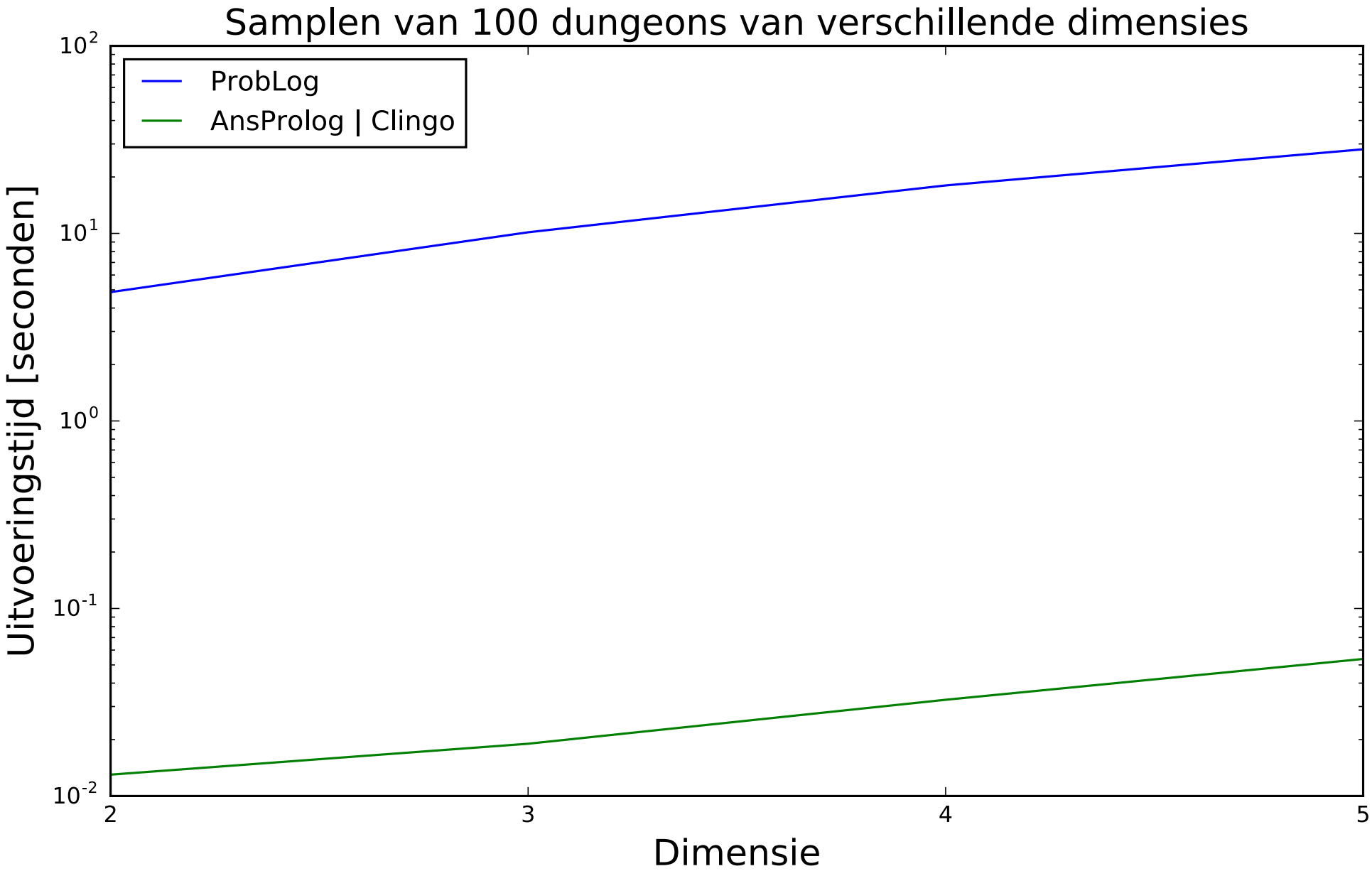
Analyse

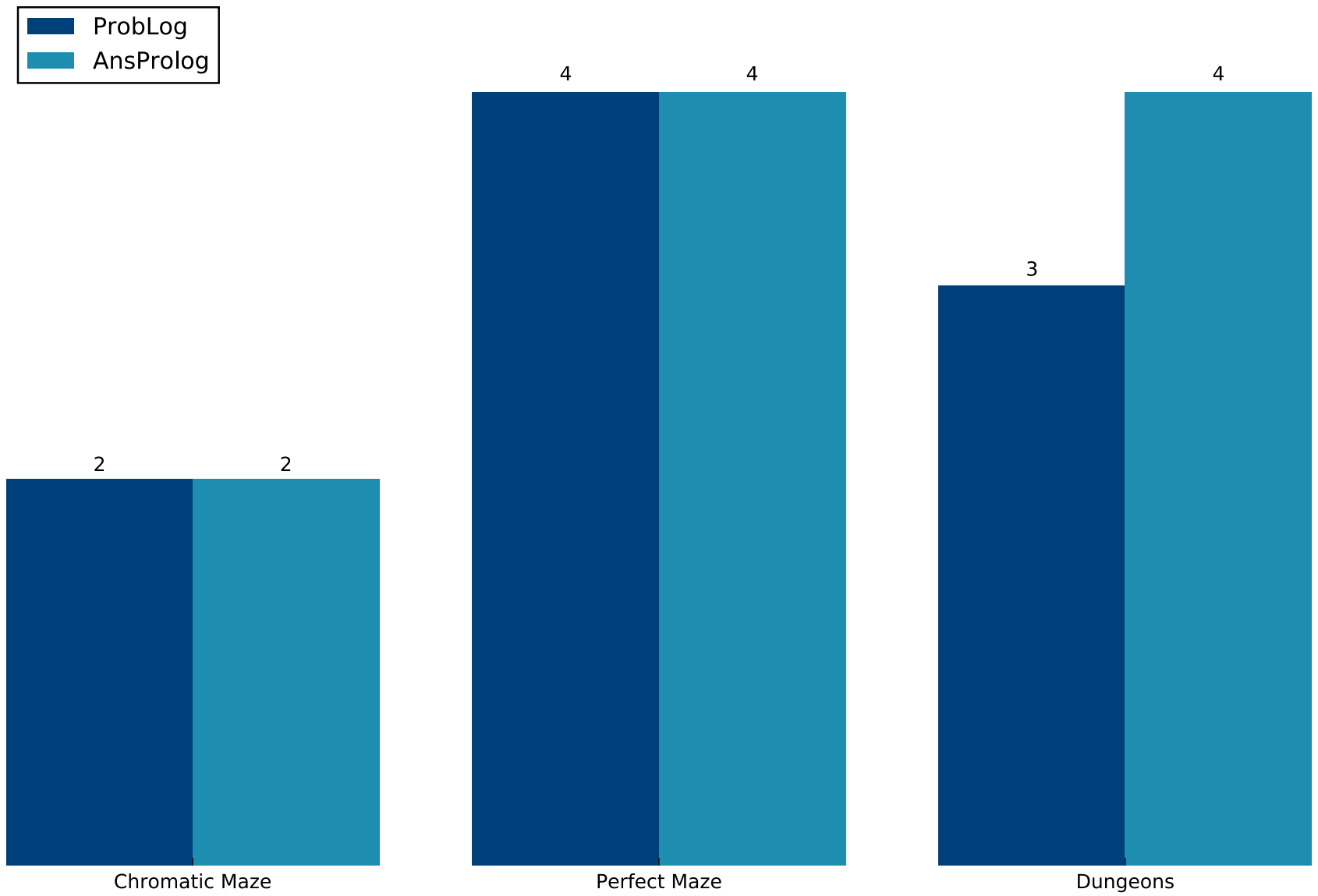
Algemeen

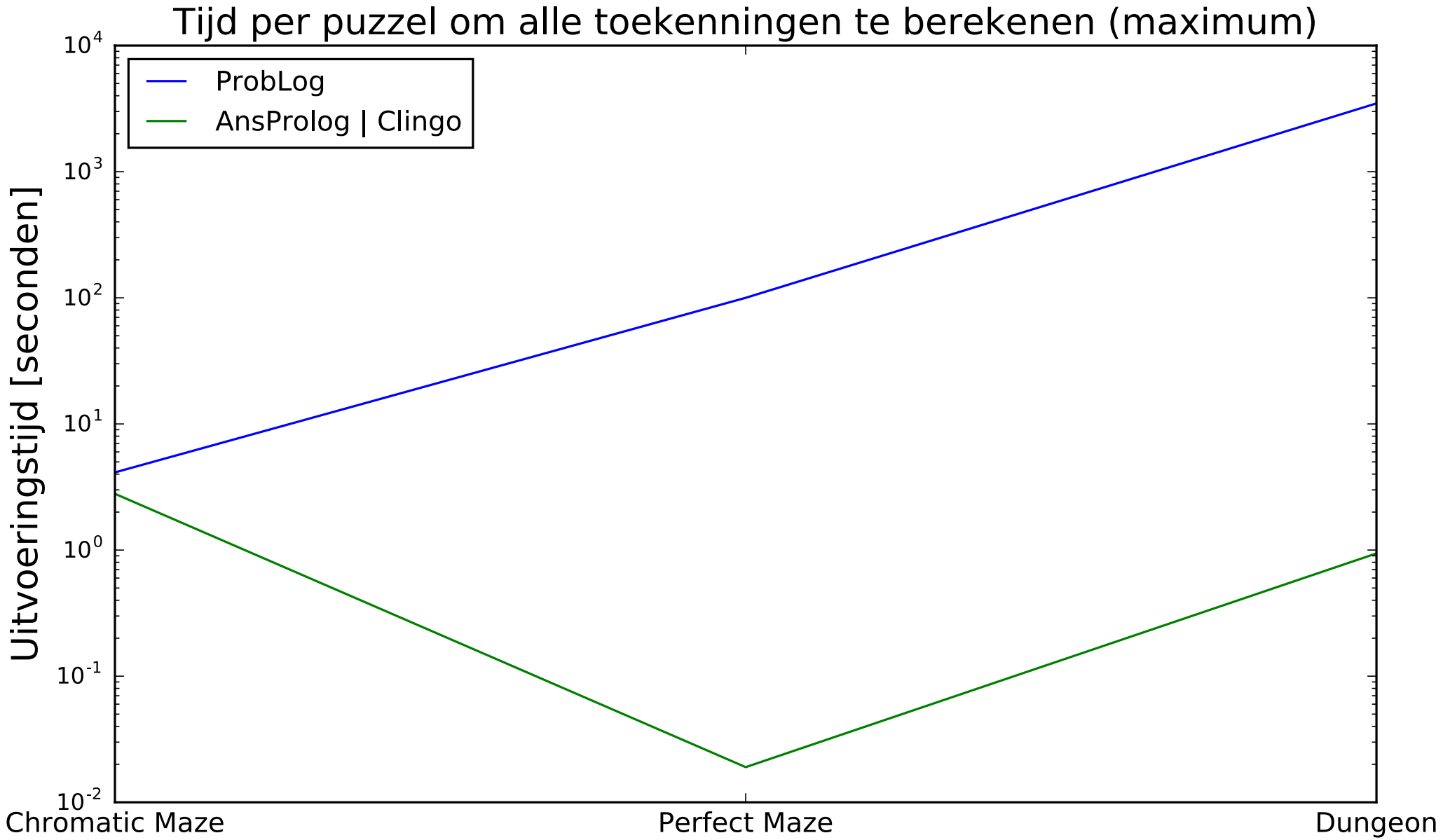
- 100 samples
- Verschillende dimensies
- Vergelijken van met en zonder evidence
- Syntactische analyse

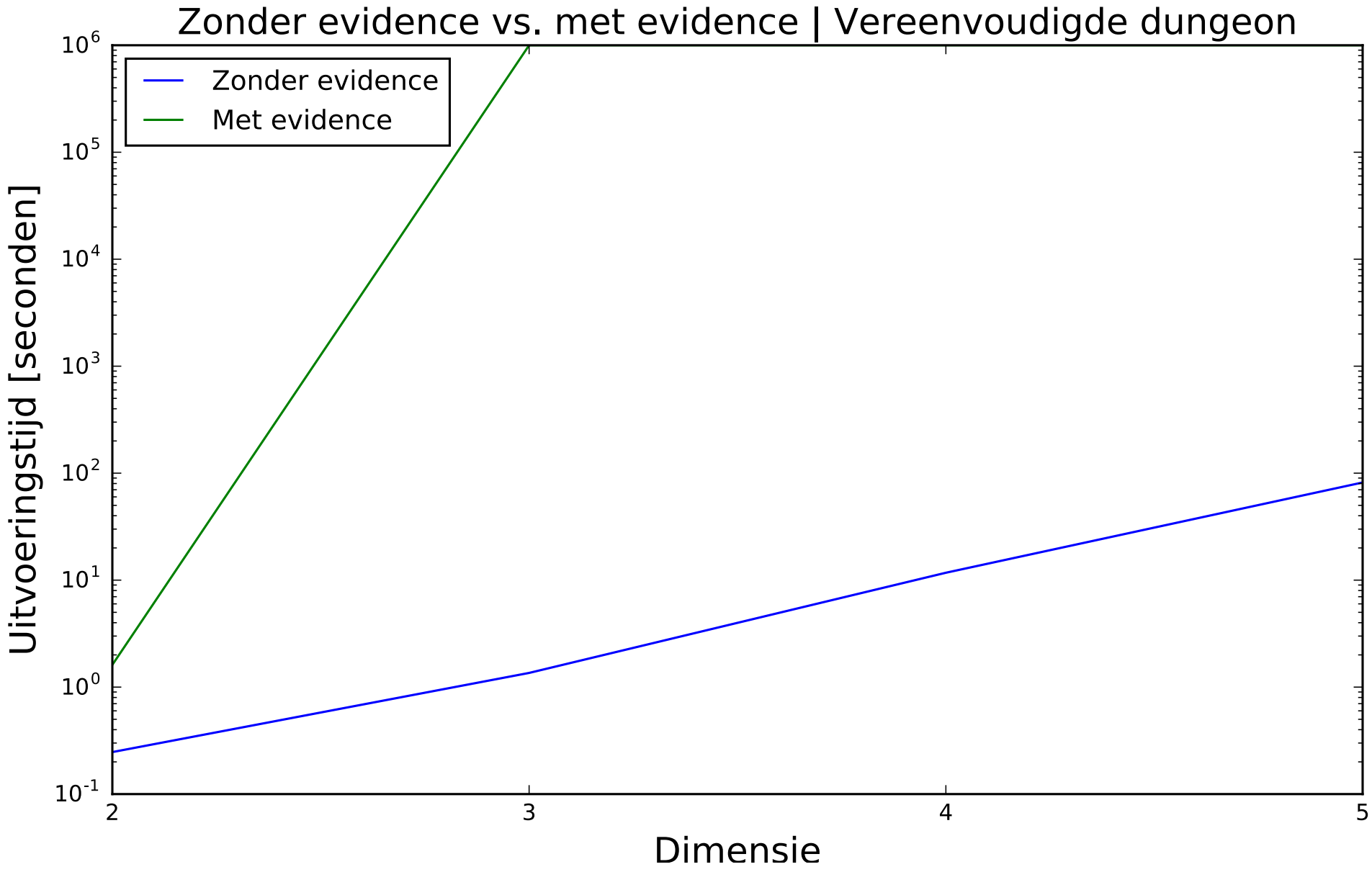

Samplen van 100 chromatic mazes van verschillende dimensies
ProbLog
AnsProlog | Clingo
Uitvoeringstijd [seconden]
10^3
10^2
10^1
10^0
10^{-1}
2
3
4
5
Dimensie


Samplen van 100 perfect mazes van verschillende dimensies
ProbLog
AnsProlog | Clingo
Uitvoeringstijd [seconden]
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
2
3
4
5
Dimensie


Samplen van 100 dungeons van verschillende dimensies
ProbLog
AnsProlog | Clingo
Uitvoeringstijd [seconden]
10^2
10^1
10^0
10^{-1}
10^{-2}
2
3
4
5
Dimensie


Hoogste dimensie met gegenereerde totaaloplosing
ProbLog
AnsProlog
2
2
4
4
3
4
Chromatic Maze
Perfect Maze
Dungeons


Tijd per puzzel om alle toekenningen te berekenen (maximum)
ProbLog
AnsProlog | Clingo
Uitvoeringstijd [seconden]
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
Chromatic Maze
Perfect Maze
Dungeon


Zonder evidence vs. met evidence | Vereenvoudigde dungeon
Zonder evidence
Met evidence
Uitvoeringstijd [seconden]
10^{6}
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
2
3
4
5
Dimensie

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Conclusie

- Sampling methode traag
- Evidence niet uitermate geschikt voor game content generation
  - Kleine kansen
  - Te veel mogelijke toekenningen voor rejection based sampling
- Evidence proberen rechtstreeks te verwerken in programma
  - Verlies van ’aanpasbaarheid’ en leesbaarheid
  - Verhoogt de complexiteit van het opstellen van programma’s enorm
- Specifiek per probleem

Overzicht

- Situering
  - Generatie van spelinhoud
  - AnsProlog
  - ProbLog
- Probleemstelling
- Motivatie
- Onderzoek
  - Code omzetten
  - Analyse
- Conclusie
- Verdere uitbreiding

Verder uitbreidingen

- Variatie onderzoeken tussen puzzels (nu mee bezig)
- Modelleringsvoor- en nadelen bekijken
- Voor- en nadelen probabiliteit goed definiëren
- Programma's herschrijven
  - Constraints kunnen niet voorkomen (soort van cut simuleren)
- Solver/grounder herschrijven (C++ ?)
- Rejection based sampling → alternatieve aanpak
  - M.a.w. betere sample methode maken

Bedankt voor jullie aandacht

Zijn er nog vragen?

KU LEUVEN