Conformant Planning as a Benchmark for QBF-Solvers

Martin Kronegger

Joint work with Andreas Pfandler and Reinhard Pichler

Institute of Information Systems
Vienna University of Technology

July 9, 2013

Dungeon Benchmark

Link

Dungeon Benchmark

Pool 1

Pool 2

⋮

Link

Dungeon Benchmark

Pool 1

Pool 2

⋮

Link

Witch

Dungeon Benchmark

Pool 1

Pool 2

⋮

Link

Witch

Dungeon

Dungeon Benchmark

Pool 1

Pool 2

⋮

Witch

Link

Dungeon

Dungeon Benchmark

Pool 1

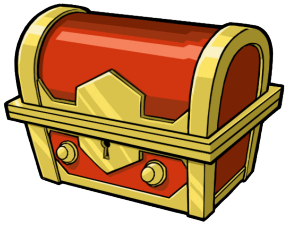
Pool 2

⋮

Witch

Link

Dungeon



Dungeon Benchmark

Pool 1

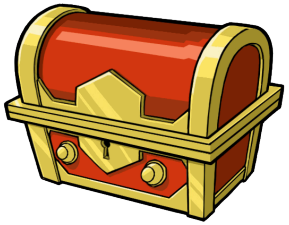
Pool 2

⋮

Witch

Link

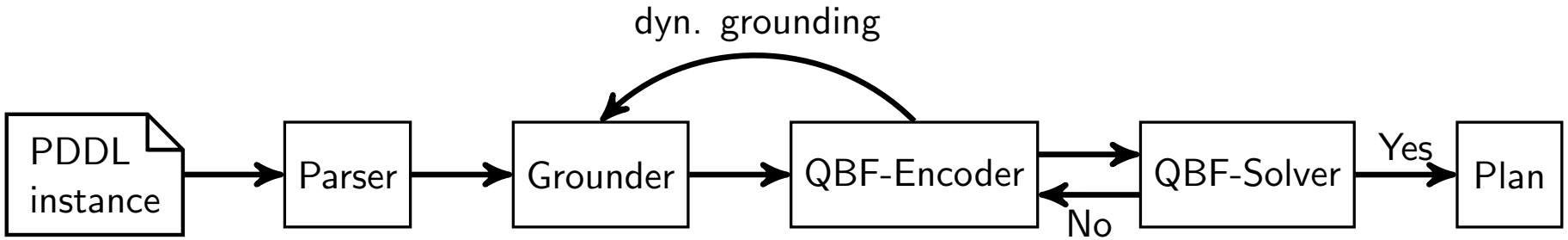
Dungeon



Parameters

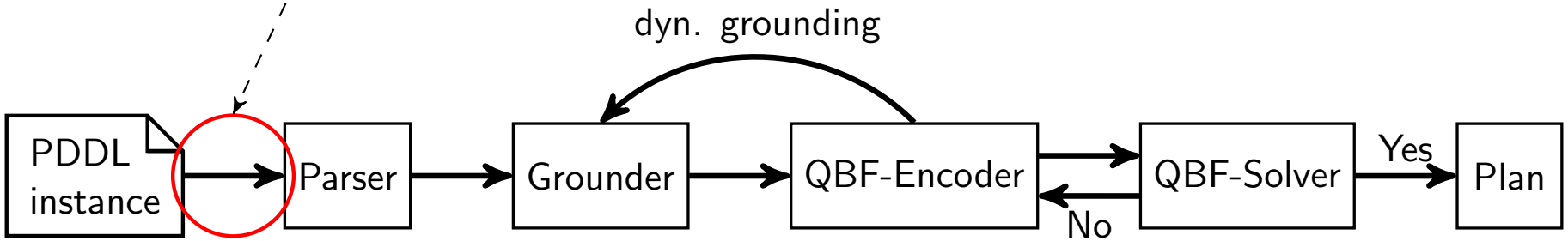
- Number of items, special items, unknowns, pools, monsters, etc.
- Min./max. size of pools, recipes, requirements for monsters, etc.
- Probability of variables to be negated (items, special items, unknowns)

Architecture of the Solver



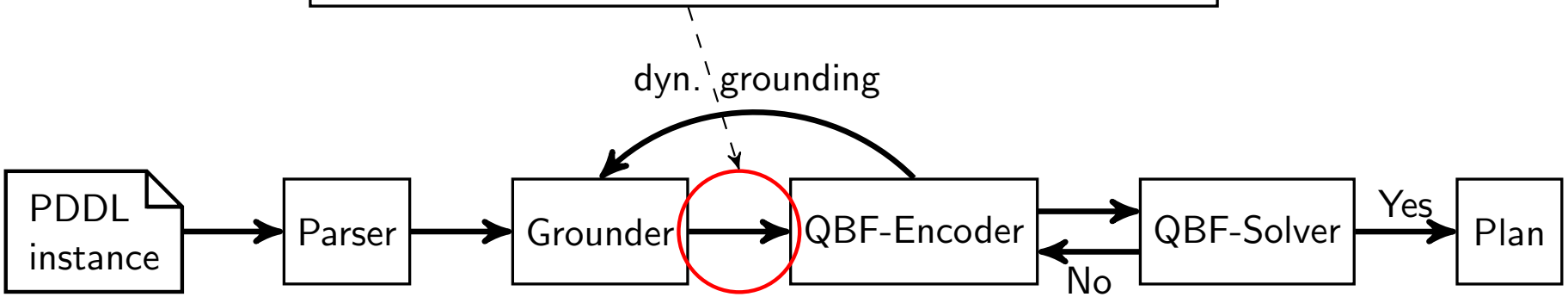
Architecture of the Solver

```
(:init (inPool i1 pool1)
       (inPool i4 pool1)
       (inPool i2 pool2)
       (unknown (in u1)) ···
)
(:goal (win g) )
(:action pick
  :parameters (?i - item ?p - pool)
  :precondition (and (allowPick ?p) (inPool ?i ?p))
  :effect (and (in ?i) (not (allowPick ?p)))
)
(:action fight ···
  :effect (and
          (when (and (in i2) (not (in u3)) (in s1)) (win g))
          (when (and (not (in i4)) (in u1)) (win g))
```



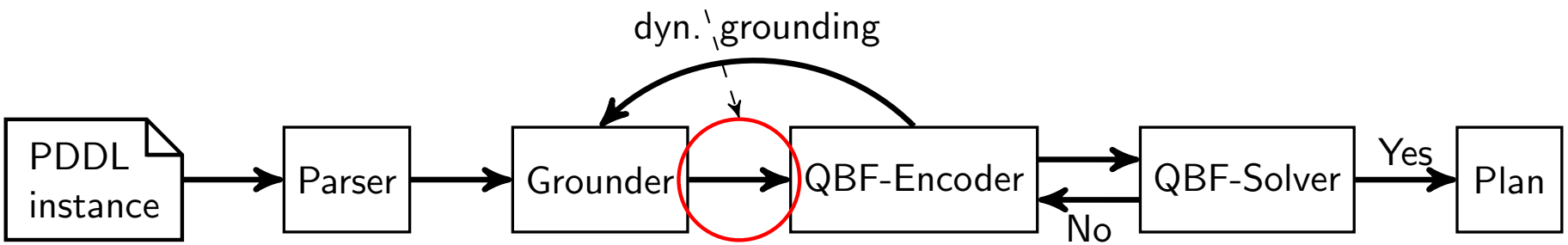
Architecture of the Solver

```
(:action pick
  :precondition (and (allowPick pool1) (inPool i1 pool1))
  :effect (and (in i1) (not (allowPick pool1)))
)
(:action pick
  :precondition (and (allowPick pool2) (inPool i1 pool2))
  :effect (and (in i1) (not (allowPick pool2)))
)
...
(:action pick
  :precondition (and (allowPick pool1) (inPool i2 pool1))
  :effect (and (in i2) (not (allowPick pool1)))
)
...
```



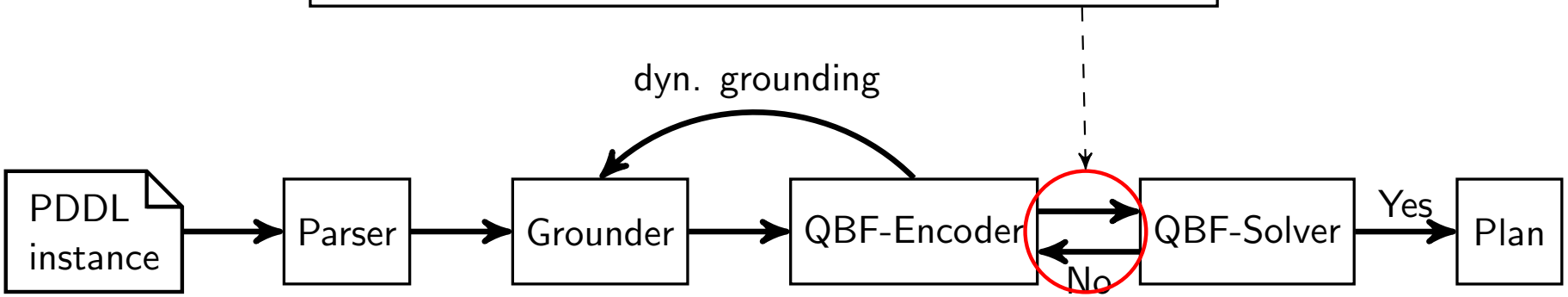
Architecture of the Solver

```
(:action pick
  :precondition (and (allowPick pool1) (inPool i1 pe
  :effect (and (in i1) (not (allowPick pool1)))
)
(:action pick
  :precondition (and (allowPick pool2) (inPool i1 pool2))
  :effect (and (in i1) (not (allowPick pool2)))
)
...
(:action pick
  :precondition (and (allowPick pool1) (inPool i2 pool1))
  :effect (and (in i2) (not (allowPick pool1)))
)
...
```



Architecture of the Solver

```
p cnf 1666 4833
e 516 518 512 514 515 1024 1025 1026 1028 1536 . . . 0
a 34 118 101 130 46 . . . 0
e 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 . . . 0
1 0
. . .
-30 0
. . .
-139 140 0
-139 -141 0
-177 46 34 130 -118 178 0
-101 193 0
-348 189 190 191 -192 349 0
-348 152 -193 349 0
. . .
```



Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

$$\psi := \exists \text{ actions } \forall \text{ unknowns } \exists \text{ helpers } \varphi$$

Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

$$\psi := \exists \text{ actions } \forall \text{ unknowns } \exists \text{ helpers } \varphi$$

$$\varphi := \underbrace{\bigwedge_{v \in \mathcal{PI}} v^0 \wedge \bigwedge_{v \in \mathcal{NI}} \neg v^0}_{\text{initial state}} \wedge \underbrace{\bigwedge_{v \in \mathcal{PG}} v^k \wedge \bigwedge_{v \in \mathcal{NG}} \neg v^k}_{\text{goal}}$$

Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

$$\psi := \exists \text{ actions } \forall \text{ unknowns } \exists \text{ helpers } \varphi$$

$$\varphi := \underbrace{\bigwedge_{v \in \mathcal{PI}} v^0 \wedge \bigwedge_{v \in \mathcal{NI}} \neg v^0}_{\text{initial state}} \wedge \underbrace{\bigwedge_{v \in \mathcal{PG}} v^k \wedge \bigwedge_{v \in \mathcal{NG}} \neg v^k}_{\text{goal}}$$

$$\wedge \bigwedge_{t \in [k]} \bigwedge_{a \in \mathcal{A}^{t-1}} \left[\left(a^{t-1} \wedge \bigwedge_{p \in \text{pre}(a)} p^{t-1} \right) \supset \bigwedge_{e \in \text{eff}(a)} e^t \right]$$

Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

$$\psi := \exists \text{ actions } \forall \text{ unknowns } \exists \text{ helpers } \varphi$$

$$\varphi := \underbrace{\bigwedge_{v \in \mathcal{PI}} v^0 \wedge \bigwedge_{v \in \mathcal{NI}} \neg v^0}_{\text{initial state}} \wedge \underbrace{\bigwedge_{v \in \mathcal{PG}} v^k \wedge \bigwedge_{v \in \mathcal{NG}} \neg v^k}_{\text{goal}}$$

$$\wedge \bigwedge_{t \in [k]} \bigwedge_{a \in \mathcal{A}^{t-1}} \left[\left(a^{t-1} \wedge \bigwedge_{p \in \text{pre}(a)} p^{t-1} \right) \supset \bigwedge_{e \in \text{eff}(a)} e^t \right]$$

$\wedge$ one action at a time

Encoding without Optimization

based on Rintanen [2007]

Goal: A quantified Boolean formula ψ that is true iff the corresponding planning problem has a plan of length k.

$$\psi := \exists \text{ actions } \forall \text{ unknowns } \exists \text{ helpers } \varphi$$

$$\varphi := \underbrace{\bigwedge_{v \in \mathcal{PI}} v^0 \wedge \bigwedge_{v \in \mathcal{NI}} \neg v^0}_{\text{initial state}} \wedge \underbrace{\bigwedge_{v \in \mathcal{PG}} v^k \wedge \bigwedge_{v \in \mathcal{NG}} \neg v^k}_{\text{goal}}$$

$$\wedge \bigwedge_{t \in [k]} \bigwedge_{a \in \mathcal{A}^{t-1}} \left[\left(a^{t-1} \wedge \bigwedge_{p \in \mathrm{pre}(a)} p^{t-1} \right) \supset \bigwedge_{e \in \mathrm{eff}(a)} e^t \right]$$

$\wedge$ one action at a time

$\wedge$ every change has a cause (framing axioms)

Advantages of this System and Benchmark Environment

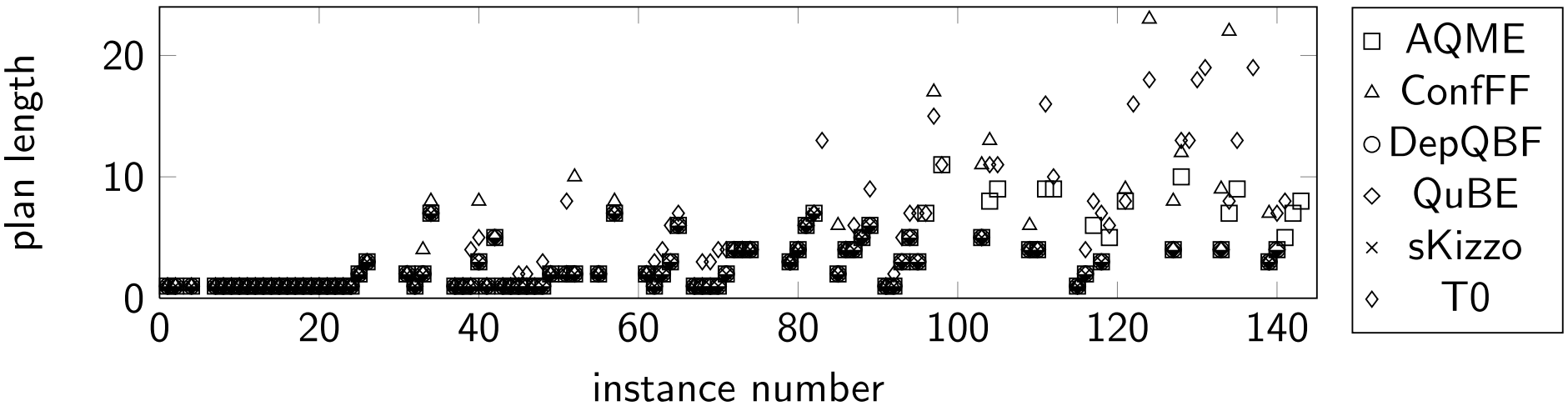
Main advantages

- Exact solver (if plan is found, then the plan has minimal length)
- Conformant planning benchmarks accessible to QBF-solvers

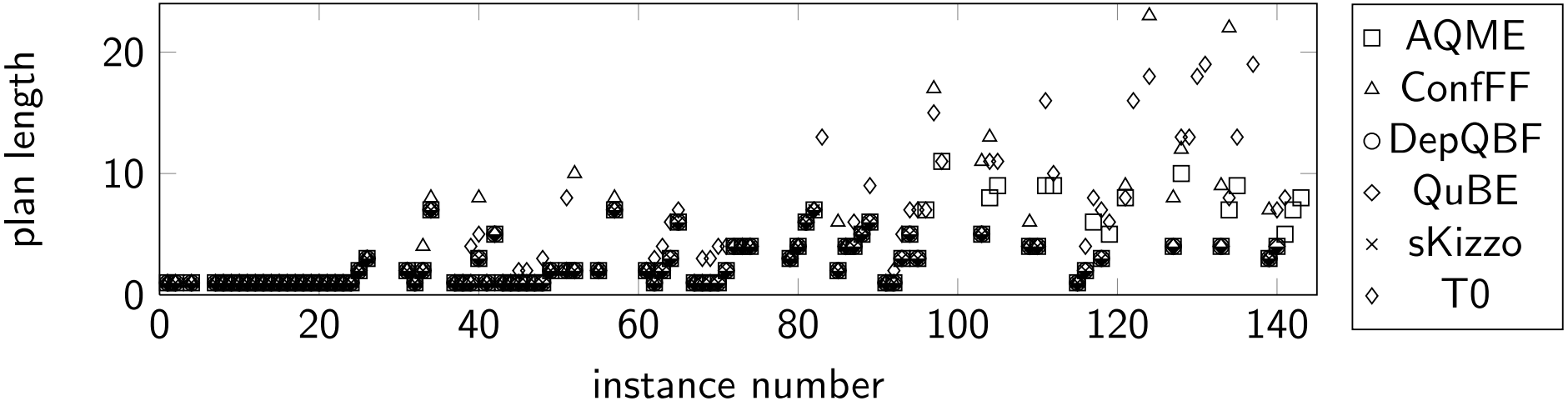
Benchmark environment

- Randomly generated Dungeon instances
- Different QBF-solvers (AQME, DepQBF, QuBE, sKizzo)
- Compare to other planning tools (ConformantFF, T0)

Benchmark Results



Benchmark Results



#	#i	#m	#u	AQME t/pl	AQME cl/vars	ConfFF t/pl	DepQBF t/pl	DepQBF cl/vars	QuBE t/pl	QuBE cl/vars	sKizzo t/pl	sKizzo cl/vars	T0 t/pl
30	10	5	10	603.53	111887	0.04	to	27478	to	9031	to	11826	0.43
				>200	41377	unsol	≥49	10271	≥16	3473	≥21	4503	unsol?
40	10	20	4	2.24	1733	437.39	0.25	1733	0.29	1733	0.32	1733	0.09
				3	648	8	3	648	3	648	3	648	5
72	15	75	10	428.26	7195	to	1.06	7195	0.92	7195	-	5398	565.85
				4	2093	-	4	2093	4	2093	ex250	1651	4
143	30	150	5	690.02	133543	to	to	65899	to	82810	-	48988	to
				8	10616	-	≥4	5756	≥5	6971	ex250	4541	-
solved instances (out of 144)				96		64	80		81		76		102

Conclusion

- System for planning with uncertainty in the initial state
- Conformant planning benchmarks are accessible to QBF-solvers
- Approach sometimes competitive to state-of-the-art planners

Conclusion

- System for planning with uncertainty in the initial state
- Conformant planning benchmarks are accessible to QBF-solvers
- Approach sometimes competitive to state-of-the-art planners

Wishlist for further improvements

- Certificates (or at least assignment to outermost quantifier-block)
- Incremental QBF-solvers
- QBF-solvers biased towards unsatisfiability

Conclusion

- System for planning with uncertainty in the initial state
- Conformant planning benchmarks are accessible to QBF-solvers
- Approach sometimes competitive to state-of-the-art planners

Wishlist for further improvements

- Certificates (or at least assignment to outermost quantifier-block)
- Incremental QBF-solvers
- QBF-solvers biased towards unsatisfiability

Future work

- More detailed experiments
- Experiments with other planning domains
- Solver specific encodings
- Improving dynamic grounding