Learning and Exploiting Progress States in Greedy Best-First Search

Patrick Ferber  Liat Cohen  Jendrik Seipp  Thomas Keller

31th International Joint Conference on Artificial Intelligence
July, 2022
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Greedy Best-First Search
Progress States
Contribution

Learn DNF formulas which identify progress states

Exploit progress state knowledge during search
Framework

\( \Pi \) expand

\( \vdash A_1 \land A_2 \land \ldots \) \lor \( (B_1 \land B_2 \land \ldots) \) \lor \ldots

\{\{\text{on}\}\} > 0, \{\{\text{clear}\}\} > 0 \ldots

\{\{\text{on}\}\} \land \{\{\text{clear}\}\} \land \ldots

\rightarrow \quad \text{learn}

---

Introduction

Learning

Experiments

Summary
Formula for $h^+$ in Gripper

$$(\lvert at_s(\cdot, \text{roomA})\rvert = 0) \lor$$
$$(\lvert at-robby_s(\text{roomA})\rvert > 0) \land (\lvert free_s(\cdot)\rvert > 0)) \lor$$
$$(\lvert at-robby_s(\text{roomB})\rvert > 0) \land (\lvert carry_s(\cdot)\rvert > 0))$$
## DNF Quality for $h^{FF}$ (F1 score)

<table>
<thead>
<tr>
<th>Domain</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Barman</strong></td>
<td>79</td>
<td>80</td>
<td><strong>81</strong></td>
<td>79</td>
<td>77</td>
</tr>
<tr>
<td><strong>Blocksworld</strong></td>
<td><strong>83</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Childsnack</strong></td>
<td>70</td>
<td>65</td>
<td>73</td>
<td><strong>81</strong></td>
<td>72</td>
</tr>
<tr>
<td><strong>Driverlog</strong></td>
<td>81</td>
<td>81</td>
<td><strong>89</strong></td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td><strong>Floortile</strong></td>
<td>75</td>
<td>85</td>
<td>85</td>
<td>85</td>
<td><strong>88</strong></td>
</tr>
<tr>
<td><strong>Gripper</strong></td>
<td>96</td>
<td>96</td>
<td>96</td>
<td><strong>98</strong></td>
<td>98</td>
</tr>
<tr>
<td><strong>Miconic</strong></td>
<td>98</td>
<td>98</td>
<td>97</td>
<td>98</td>
<td><strong>99</strong></td>
</tr>
<tr>
<td><strong>Visitall</strong></td>
<td>70</td>
<td>70</td>
<td><strong>73</strong></td>
<td>73</td>
<td>73</td>
</tr>
</tbody>
</table>
Use Case: Tie-Breaking
Expansions

$h_{L*}^F$ (lower for 215 tasks)
$h_{L*}^F$ (lower for 43 tasks)
$h^+ (lower for 51 tasks)$
$h^+ (lower for 7 tasks)$
Learning and Exploiting Progress States in Greedy Best-First Search
Patrick Ferber, Liat Cohen, Jendrik Seipp, Thomas Keller

Identify Progress States
- DNF formula
- Description Logic features

Use Case for Progress States
- Reduce expansions
- Often increases runtime
Learning and Exploiting Progress States in Greedy Best-First Search
Patrick Ferber, Liat Cohen, Jendrik Seipp, Thomas Keller

Identify Progress States
- DNF formula
- Description Logic features

Use Case for Progress States
- Reduce expansions
- Often increases runtime
High-Water Mark
High-Water Mark
High-Water Mark
High-Water Mark
High-Water Mark
## Runtime

<table>
<thead>
<tr>
<th>Domain</th>
<th>$h^{FF}$</th>
<th>+DNF</th>
<th>$h^+$</th>
<th>+DNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barman</td>
<td>3.4</td>
<td>+462%</td>
<td>161</td>
<td>-4%</td>
</tr>
<tr>
<td>Blocksworld</td>
<td>0.3</td>
<td>+67%</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Childsnack</td>
<td>3.5</td>
<td>+94%</td>
<td>88</td>
<td>-1%</td>
</tr>
<tr>
<td>Driverlog</td>
<td>0.1</td>
<td>+3100%</td>
<td>244</td>
<td>+3%</td>
</tr>
<tr>
<td>Floortile</td>
<td>0.4</td>
<td>+125%</td>
<td>11</td>
<td>-0%</td>
</tr>
<tr>
<td>Gripper</td>
<td>1.1</td>
<td>-45%</td>
<td>301</td>
<td>-12%</td>
</tr>
<tr>
<td>Miconic</td>
<td>0.2</td>
<td>+650%</td>
<td>103</td>
<td>-8%</td>
</tr>
<tr>
<td>Visitall</td>
<td>0.0</td>
<td>+10700%</td>
<td>305</td>
<td>-1%</td>
</tr>
</tbody>
</table>