Planejamento sob Incerteza para Metas de Alcançabilidade Estendidas

Silvio do Lago Pereira & Leliane Nunes de Barros

DCC-IME-USP

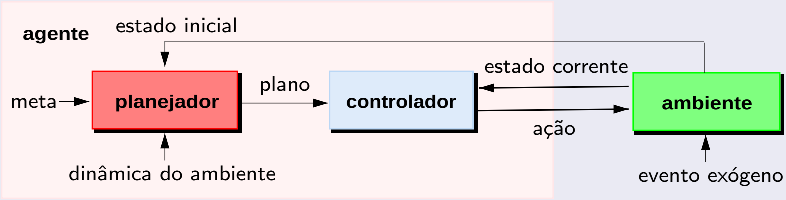
Sumário

Planejamento automatizado

Planejamento sob incerteza

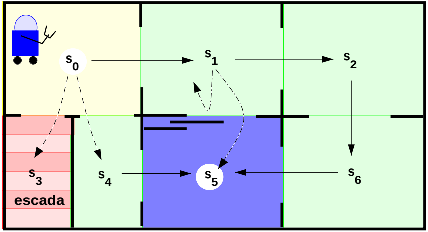
Planejamento Automatizado é uma área da IA que visa desenvolver algoritmos para sintetizar planos a partir da análise de uma descrição formal da dinâmica do ambiente, do estado inicial e da meta do agente.

Planejamento sob incerteza (plano ~ política de comportamento)



Domínios de planejamento

Problemas e qualidades de solução

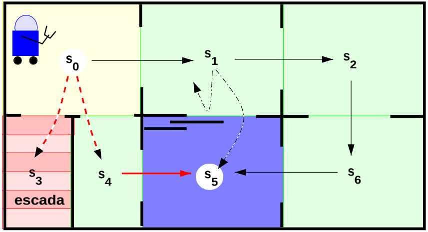


Qualidades de soluções: fraca, forte-cíclica e forte

- Que garantia de alcançar a meta uma solução particular oferece?

Domínios de planejamento

Problemas e qualidades de solução

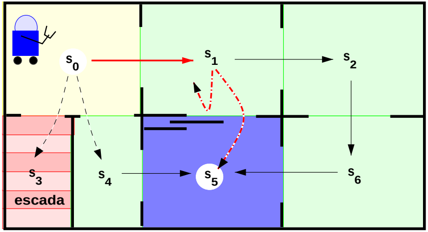


Qualidades de soluções: fraca, forte-cíclica e forte

- Que garantia de alcançar a meta uma solução particular oferece?
- $\pi_1 = \{(s_0, \textit{entrar-em-}s_4), (s_4, \textit{entrar-em-}s_5)\}$

Domínios de planejamento

Problemas e qualidades de solução

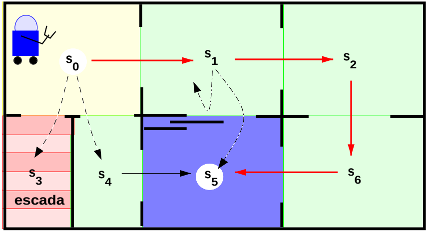


Qualidades de soluções: fraca, forte-cíclica e forte

- Que garantia de alcançar a meta uma solução particular oferece?
- $\pi_1 = \{(s_0, \mathit{entrar\text{-}em\text{-}s_4}), (s_4, \mathit{entrar\text{-}em\text{-}s_5})\}$
- $\pi_2 = \{(s_0, \mathit{entrar\text{-}em\text{-}s_1}), (s_1, \mathit{entrar\text{-}em\text{-}s_5})\}$

Domínios de planejamento

Problemas e qualidades de solução

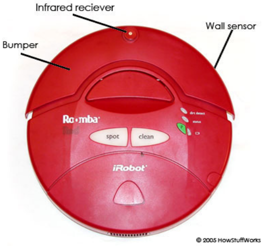


Qualidades de soluções: fraca, forte-cíclica e forte

- Que garantia de alcançar a meta uma solução particular oferece?
- $\pi_1 = \{(s_0, \textit{entrar-em-}s_4), (s_4, \textit{entrar-em-}s_5)\}$
- $\pi_2 = \{(s_0, \textit{entrar-em-}s_1), (s_1, \textit{entrar-em-}s_5)\}$
- $\pi_3 = \{(s_0, \textit{entrar-em-}s_1), (s_1, \textit{entrar-em-}s_2), (s_2, \textit{entrar-em-}s_6), (s_6, \textit{entrar-em-}s_5)\}$

Sobre esse trabalho
Motivação e objetivos

- IPC-2006: "*The competition will focus only on planning for goal reachability*"
- estado-da-arte ainda está voltado para metas com expressividade limitada
- porém, metas mais expressivas têm despertado grande interesse na área



Objetivos

- tratar problemas de planejamento para metas mais expressivas
- usar um método formal para garantir a qualidade das soluções
- sintetizar planos usando técnicas de verificação de modelos

Verificação de Modelos

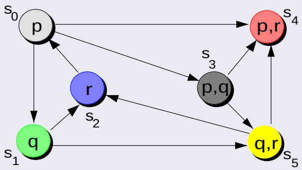
O arcabouço

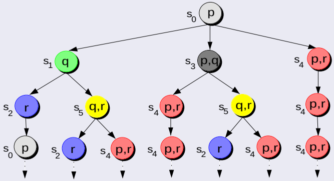
Verificação de modelos

consiste em decidir se $\mathcal{K} \models \varphi$, onde:

- $\mathcal{K}$ é um modelo formal do sistema (**estrutura de Kripke**)
- φ é uma propriedade a ser verificada (**fórmula de lógica temporal**)

Estrutura de Kripke $\mathcal{K} = \langle \mathcal{S}, \mathcal{L}, \mathcal{T} \rangle$ com assinatura $\mathbb{P}$ e árvore de computação





Verificação de Modelos

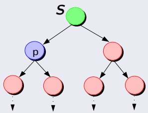
CTL - *Computation Tree Logic*

Operadores temporais

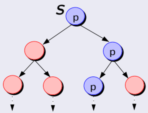
- $\bigcirc$: sucessor
- $\square$: invariante
- $\diamond$: finalmente
- $\sqcup$: até que

Semântica

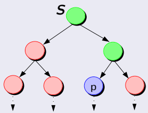
$s \models \exists \bigcirc p$



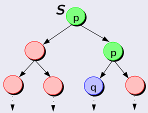
$s \models \exists \square p$



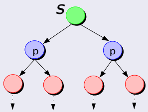
$s \models \exists \diamond p$



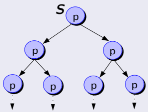
$s \models \exists (p \sqcup q)$



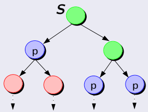
$s \models \forall \bigcirc p$



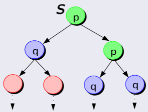
$s \models \forall \square p$



$s \models \forall \diamond p$



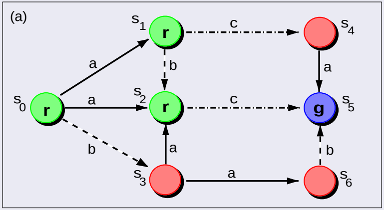
$s \models \forall (p \sqcup q)$



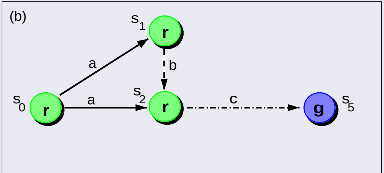
Planejamento baseado em Verificação de Modelos

Domínios, políticas e estruturas

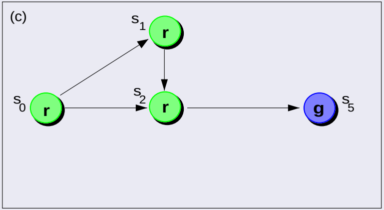
Domínio de planejamento $\mathcal{D} = \langle \mathcal{S}, \mathcal{L}, \mathcal{T} \rangle$ com assinatura $(\mathbb{P}, \mathbb{A})$



Estrutura de execução $\mathcal{D}_\pi$ e estrutura de Kripke correspondente $\mathcal{K}(\mathcal{D}_\pi)$



$\pi_1 = \{(s_0, a), (s_1, b), (s_2, c)\}$



Planejamento baseado em Verificação de Modelos

Problemas, soluções e validação

Um problema de planejamento para meta de alcançabilidade simples

é definido por uma tupla $\mathcal{P} = \langle \mathcal{D}, s_0, \phi \rangle$, onde:

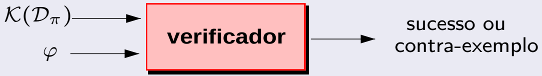
- $\mathcal{D} = \langle \mathcal{S}, \mathcal{L}, \mathcal{T} \rangle$ é um domínio com assinatura $(\mathbb{P}, \mathbb{A})$
- $s_0 \in \mathcal{S}$ é o estado inicial do ambiente
- ϕ é uma fórmula proposicional sobre $\mathbb{P}$

Caracterização das classes de soluções em CTL

Seja π uma política para $\mathcal{P} = \langle \mathcal{D}, s_0, \phi \rangle$. Então, π é uma solução:

- **fraca** para $\mathcal{P}$ $\Leftrightarrow (\mathcal{K}(\mathcal{D}_\pi), s_0) \models \exists \Diamond \phi$
- **forte** para $\mathcal{P}$ $\Leftrightarrow (\mathcal{K}(\mathcal{D}_\pi), s_0) \models \forall \Diamond \phi$
- **forte-cíclica** para $\mathcal{P}$ $\Leftrightarrow (\mathcal{K}(\mathcal{D}_\pi), s_0) \models \forall \Box \exists \Diamond \phi$

Validação de políticas usando CTL



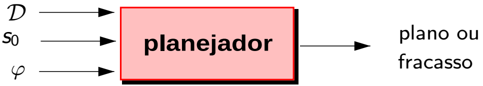
Nosso propósito

Síntese como efeito colateral da verificação de modelos

Planejamento como verificação de modelos

consiste em decidir se $(\mathcal{D}, s_0) \models \varphi$, onde:

- $\mathcal{D}$ é um modelo do ambiente de planejamento
- s_0 é o estado inicial do ambiente
- φ é uma **meta de alcançabilidade estendida**



Uma meta de alcançabilidade estendida é um par (φ_1, φ_2), onde:

- φ_1 é uma condição a ser preservada
- φ_2 é uma condição a ser alcançada

Nosso propósito

Planejamento para metas de alcançabilidade estendidas

Subclasses de metas de alcançabilidade estendidas da forma (φ_1, φ_2)

- simples $\Leftrightarrow$ φ_1 é a constante $\top$
- linear $\Leftrightarrow$ φ_1 é uma fórmula proposicional
- ramificada $\Leftrightarrow$ φ_1 é uma fórmula temporal

alcançabilidade estendida

alcançabilidade estendida ramificada

alcançabilidade estendida linear

alcançabilidade simples

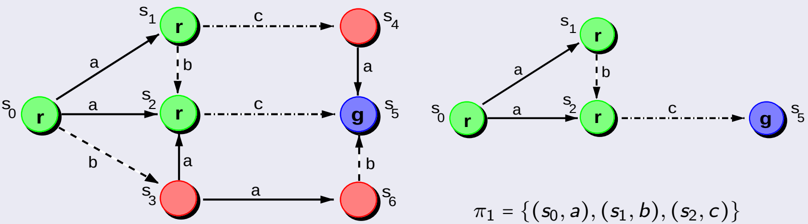
CTL x metas de alcançabilidade estendidas

tarefa	**linear**	**ramificada**
especificação	sim ✓	não ✗
síntese	não ✗	não ✗
validação	sim ✓	não ✗

Inadequação de CTL para o nosso propósito

Exemplo 1 - Problema com meta da subclasse linear

Domínio $\mathcal{D}_1$: alcançar g, garantidamente, preservando r



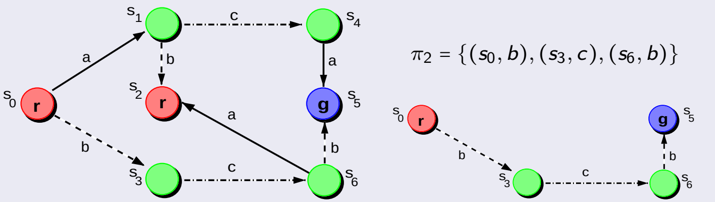
$\pi_1 = \{(s_0, a), (s_1, b), (s_2, c)\}$

- ✓ Especificação: $\forall(r \sqcup g)$
- X Síntese: $(\mathcal{K}(\mathcal{D}^1), s_0) \nvDash \forall(r \sqcup g)$
- ✓ Validação: $(\mathcal{K}(\mathcal{D}_1^{\pi_1}), s_0) \vDash \forall(r \sqcup g)$

Inadequação de CTL para o nosso propósito

Exemplo 2 - Problema com meta da subclasse ramificada

Domínio $\mathcal{D}_2$: alcançar g, garantidamente, preservando a propriedade de sempre poder alcançar r em no máximo dois passos



- X Especificação: $\forall((r \vee \forall \bigcirc r \vee \forall \bigcirc \forall \bigcirc r) \sqcup g)$
- X Síntese: $(\mathcal{K}(\mathcal{D}^2), s_0) \not\models \forall((r \vee \forall \bigcirc r \vee \forall \bigcirc \forall \bigcirc r) \sqcup g)$
- X Validação: $(\mathcal{K}(\mathcal{D}_2^{\pi_2}), s_0) \not\models \forall((r \vee \forall \bigcirc r \vee \forall \bigcirc \forall \bigcirc r) \sqcup g)$

Nem toda política pode ser validada após ter sido sintetizada!

A lógica temporal α-CTL

Sintaxe e semântica

Novos operadores temporais [JAAMAS-2008]

- CTL: $\boxed{\forall \bigcirc p}$ vale em s se todo sucessor de s satisfaz p
- α-CTL: $\boxed{\forall \odot p}$ vale em s se todo α-sucessor de s, para $\alpha \in \mathbb{A}$, satisfaz p

Intensão das fórmulas

- $[\![p]\!]_{\mathcal{D}} = \{s \in \mathcal{S} : p \in \mathcal{L}(s)\}$
- $[\![\neg p]\!]_{\mathcal{D}} = \mathcal{S} \setminus [\![p]\!]_{\mathcal{D}}$
- $[\![\varphi_1 \wedge \varphi_2]\!]_{\mathcal{D}} = [\![\varphi_1]\!]_{\mathcal{D}} \cap [\![\varphi_2]\!]_{\mathcal{D}}$
- $[\![\varphi_1 \vee \varphi_2]\!]_{\mathcal{D}} = [\![\varphi_1]\!]_{\mathcal{D}} \cup [\![\varphi_2]\!]_{\mathcal{D}}$
- $[\![\exists \odot \varphi_1]\!]_{\mathcal{D}} = \mathcal{T}_{\exists}^{-}([\![\varphi_1]\!]_{\mathcal{D}}) = \{s \in \mathcal{S} : a \in \mathbb{A} \text{ e } \mathcal{T}(s,a) \cap Y \neq \varnothing\}$
- $[\![\forall \odot \varphi_1]\!]_{\mathcal{D}} = \mathcal{T}_{\forall}^{-}([\![\varphi_1]\!]_{\mathcal{D}}) = \{s \in \mathcal{S} : a \in \mathbb{A} \text{ e } \varnothing \neq \mathcal{T}(s,a) \subseteq Y\}$
- $[\![\exists \boxdot \varphi_1]\!]_{\mathcal{D}} = \nu Y.([\![\varphi_1]\!]_{\mathcal{D}} \cap \mathcal{T}_{\exists}^{-}(Y))$
- $[\![\forall \boxdot \varphi_1]\!]_{\mathcal{D}} = \nu Y.([\![\varphi_1]\!]_{\mathcal{D}} \cap \mathcal{T}_{\forall}^{-}(Y))$
- $[\![\exists(\varphi_1 \sqcup \varphi_2)]\!]_{\mathcal{D}} = \mu Y.([\![\varphi_2]\!]_{\mathcal{D}} \cup ([\![\varphi_1]\!]_{\mathcal{D}} \cap \mathcal{T}_{\exists}^{-}(Y)))$
- $[\![\forall(\varphi_1 \sqcup \varphi_2)]\!]_{\mathcal{D}} = \mu Y.([\![\varphi_2]\!]_{\mathcal{D}} \cup ([\![\varphi_1]\!]_{\mathcal{D}} \cap \mathcal{T}_{\forall}^{-}(Y)))$

A lógica temporal α-CTL

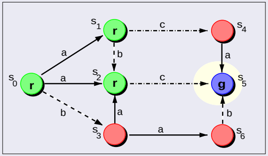
Verificação de modelos

O verificador VACTL [SBMF-2007]

VACTL$(\varphi, \mathcal{D})$
1 $C \leftarrow \mathcal{S} \setminus$ INTENSÃO$[\mathcal{D}](\varphi)$
2 se $C = \varnothing$ então devolva sucesso
3 senão devolva C

Exemplo - Verificação de $\forall(r \sqcup g)$

$\{s_5\}$



Teoremas e propriedades formais

- INTENSÃO$[\mathcal{D}](\varphi)$ devolve o conjunto $[\![\varphi]\!]_{\mathcal{D}}$
- VACTL$(\mathcal{D}, \varphi)$ devolve sucesso $\Leftrightarrow (\mathcal{D}, s) \vDash \varphi$, para $\forall s \in \mathcal{S}$

A lógica temporal α-CTL

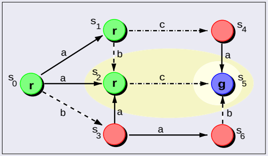
Verificação de modelos

O verificador VACTL [SBMF-2007]

$\textsc{Vactl}(\varphi, \mathcal{D})$
1 $C \leftarrow \mathcal{S} \setminus \textsc{Intensão}[\mathcal{D}](\varphi)$
2 `se` $C = \emptyset$ `então devolva sucesso`
3 `senão devolva` C

Exemplo - Verificação de $\forall(r \sqcup g)$

$\{s_5, s_2\}$



Teoremas e propriedades formais

- $\textsc{Intensão}[\mathcal{D}](\varphi)$ devolve o conjunto $[\![\varphi]\!]_{\mathcal{D}}$
- $\textsc{Vactl}(\mathcal{D}, \varphi)$ devolve sucesso $\Leftrightarrow (\mathcal{D}, s) \vDash \varphi$, para $\forall s \in \mathcal{S}$

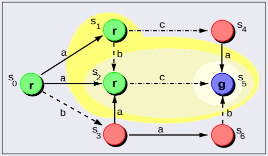
A lógica temporal α-CTL
Verificação de modelos

O verificador VACTL [SBMF-2007]

VACTL$(\varphi, \mathcal{D})$
1 $C \leftarrow \mathcal{S} \setminus$ INTENSÃO$[\mathcal{D}](\varphi)$
2 se $C = \varnothing$ então devolva sucesso
3 senão devolva C

Exemplo - Verificação de $\forall(r \sqcup g)$

$\{s_5, s_2, s_1\}$



Teoremas e propriedades formais

- INTENSÃO$[\mathcal{D}](\varphi)$ devolve o conjunto $[\![\varphi]\!]_{\mathcal{D}}$
- VACTL$(\mathcal{D}, \varphi)$ devolve sucesso $\Leftrightarrow (\mathcal{D}, s) \vDash \varphi$, para $\forall s \in \mathcal{S}$

A lógica temporal α-CTL

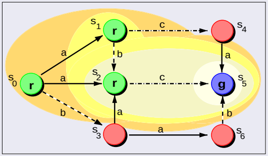
Verificação de modelos

O verificador VACTL **[SBMF-2007]**

$\textsc{Vactl}(\varphi, \mathcal{D})$
$C \leftarrow \mathcal{S} \setminus \textsc{Intensão}[\mathcal{D}](\varphi)$
`se` $C = \varnothing$ `então devolva sucesso`
`senão devolva` C

Exemplo - Verificação de $\forall(r \sqcup g)$

$\{s_5, s_2, s_1, \color{red}{s_0}\} \Leftarrow$ ponto-fixo mínimo



Teoremas e propriedades formais

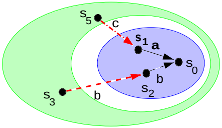
- $\textsc{Intensão}[\mathcal{D}](\varphi)$ devolve o conjunto $[\![\varphi]\!]_{\mathcal{D}}$
- $\textsc{Vactl}(\mathcal{D}, \varphi)$ devolve sucesso $\Leftrightarrow (\mathcal{D}, s) \models \varphi$, para $\forall s \in \mathcal{S}$

A lógica temporal α-CTL

Síntese de modelos

O planejador PACTL [JAAMAS-2008, JCSS-2009]

$\textsc{Pactl}(\mathcal{D}, s_0, \varphi)$
1 $M \leftarrow \textsc{Modelo}[\mathcal{D}](\texttt{min}, \varphi)$
2 $C \leftarrow \textsc{Cobertura}(M)$
3 **se** $s_0 \in C$ **então devolva** $\textsc{Política}(M)$
4 **devolva fracasso**



Teoremas e propriedades formais

- $\textsc{Pactl}(\mathcal{D}, s_0, \varphi)$ devolve fracasso $\Leftrightarrow (\mathcal{D}, s_0) \nvDash \varphi$
- $(\mathcal{D}, s_0) \vDash \exists(\varphi_1 \sqcup \varphi_2) \Rightarrow$ PACTL devolve uma solução fraca
- $(\mathcal{D}, s_0) \vDash \forall(\varphi_1 \sqcup \varphi_2) \Rightarrow$ PACTL devolve uma solução forte
- $(\mathcal{D}, s_0) \vDash \forall \boxdot \exists(\varphi_1 \sqcup \varphi_2) \Rightarrow$ PACTL devolve uma solução forte-cíclica
- A solução fraca devolvida por PACTL é ótima no melhor caso
- A solução forte devolvida por PACTL é ótima no pior caso

Planejamento baseado em α-CTL

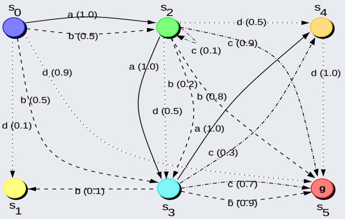
Outros resultados

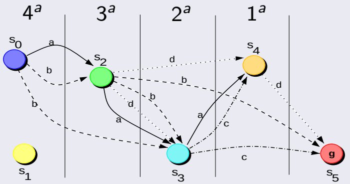
Tratamento de outros tipos de metas **[WoLLIC-2008, LJIGPL-2008]**

- comparação entre α-CTL e P-CTL
- especificação de outros tipos de metas estendidas
- planejamento para metas do tipo "*try-your-best*"

Planejamento probabilístico forte **[ICAPS-2008, MICAI-2008]**

- descarta ações que causam falhas ou ciclos (ordenação topológica)
- usa equações de Bellman para escolher melhor ação em cada etapa
- permite metas de alcançabilidade estendidas





Conclusão

Resumo e Contribuições

A síntese de planos pode ser obtida como um efeito colateral da verificação de uma propriedade φ (meta) num modelo $\mathcal{D}$ (domínio). Assim, a validade de um plano é conseqüência direta de um processo de síntese bem fundamentado em métodos formais.

Contribuições originais desse trabalho

- definição da classe de metas de alcançabilidade estendidas, que são mais expressivas que aquelas tratadas no planejamento clássico
- definição da lógica temporal α-CTL, cuja semântica permite o tratamento adequado de metas de alcançabilidade estendidas
- formulação de um arcabouço formal para planejamento sob incerteza para metas de alcançabilidade estendidas, com diferentes requisitos de qualidade (fraca, forte ou forte-cíclica)
- criação de um algoritmo para planejamento probabilístico forte para metas de alcançabilidade estendidas, que integra idéias de VM e MDPs