5
Estratégias de Busca em Teoria Local dos Conjuntos

Neste capítulo, definimos a noção de estratégia de busca e identificamos, no *topos* **ECF**, construções que podem ser usadas para representar tais estratégias. Expressamos estas contruções na lógica interna de **ECF** usando teoria local dos conjuntos ([13]).

5.1
Busca

5.1.1 Heurística = construção de florestas + busca. Tendo definido espaços de busca, devemos agora voltar nossa atenção para outros aspectos de heurísticas. Em especial, desejamos percorrer, segundo alguma estratégia bem definida, as florestas associadas aos problemas.

5.1.2 Busca sequencial em uma floresta. Fazer uma busca sequencial em uma floresta consiste, basicamente, no seguinte algoritmo:

```
1.  Visitar um nó do nível 0 da floresta
    (i.e., a raiz de alguma árvore da floresta);

2.  Repetir até alguma condição de fim:

    2a.    Visitar um nó já visitado
           ou o sucessor de um nó já visitado.
```

Este algoritmo simples será usado como base de nossas definições relativas a estratégias de busca. Note que é a escolha do próximo nó a visitar, no passo `2a`, que varia entre estratégias diferentes. Esta escolha pode se dar baseada apenas na posição dos nós já visitados e seus sucessores (como em busca em profundidade e outros métodos de busca por força bruta), baseada na qualidade dos nós já visitados e de seus sucessores (como em uma heurística gulosa e outros métodos de busca heurística), ou em uma combinação destes e de outros fatores.

5.1.3 Busca como uma sequência de sub-árvores.

Dada uma floresta T qualquer, podemos representar o comportamento de uma estratégia de busca através de uma sequência de sub-árvores de T satisfazendo certas condições. A cada iteração, os nós visitados até então formam uma sub- árvore de T, que vai sendo expandida ao longo da busca.

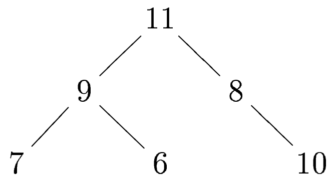
O seguinte diagrama na categoria **Floresta** representa este processo:

$$
\begin{tikzcd}
 & & T & & \\
T_1 \arrow[urr, tail, "f_1"] \arrow[r, tail, "f_{(1,2)}"'] & T_2 \arrow[ur, tail, "f_2"] \arrow[r, tail, "f_{(2,3)}"'] & T_3 \arrow[u, tail, "f_3"] \arrow[r, tail, "f_{(3,4)}"'] & T_4 \arrow[ul, tail, "f_4"'] \arrow[r, tail, "f_{(4,5)}"'] & \cdots
\end{tikzcd}
$$

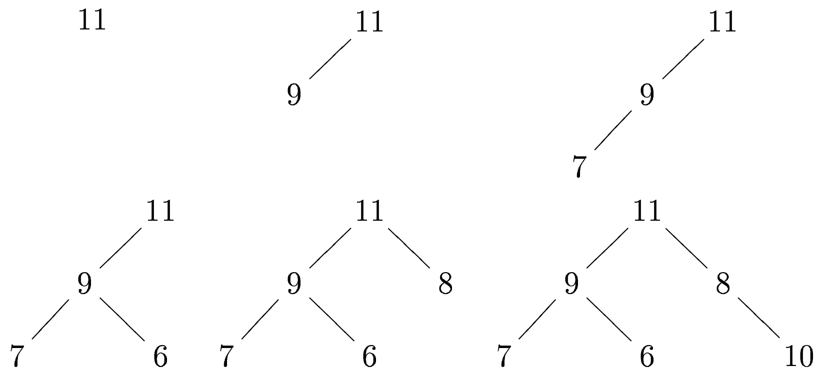
Todos os morfismos mostrados são imersões de florestas (i.e., monomorfismos). Para que o diagrama defina uma estratégia de busca sequencial na floresta T, as seguintes condições devem ser satisfeitas:

- O diagrama comuta; i.e., $f_1 = f_2 \circ f_{(1,2)}$, $f_2 = f_3 \circ f_{(2,3)}$, etc.
- O domínio T_1 de f_1 é uma floresta que consiste de um único nó; em outras palavras, f_1 escolhe o nó inicial da busca;
- Os domínios T_i de todos os morfismos f_i são árvores, o que significa que nós são visitados apenas através de arestas existentes;
- Cada árvore T_{i+1} possui no máximo um nó a mais do que a árvore T_i; este nó a mais, caso exista, é o novo nó visitado no estágio $i + 1$ da busca.

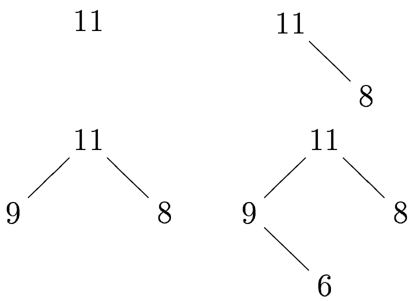
5.1.4 Exemplo. Dada a floresta abaixo (composta de uma única árvore),



uma estratégia de busca sequencial em profundidade produziria a seguinte sequência de sub-árvores:



Já uma busca *best-first* poderia produzir a seguinte sequência de sub-árvores, supondo que o número de cada nó denota sua qualidade (valores menores representando maior qualidade):



5.1.5 Visitando o mesmo nó diversas vezes. O passo 2a do algoritmo de busca sequencial em 5.1.2 permite que um nó visitado anteriormente seja visitado outras vezes durante a busca. Para acomodar esta possibilidade, é conveniente representar o comportamento de uma estratégia de busca em uma floresta T por uma sequência de pares

$$\{(T_1, v_1), (T_2, v_2), \ldots, (T_i, v_i), \ldots\}$$

onde cada T_i é uma sub-árvore de T conforme descrito em 5.1.3, e cada v_i é um nó de T_i tal que, para $i > 1$, se $T_i \neq T_{i-1}$, então v_i é o único nó presente em T_i que não está presente em T_{i-1}. Se $T_i = T_{i-1}$, então v_i é um nó que está sendo visitado outra vez.

5.2
Múltiplas Florestas

5.2.1 Busca em uma floresta × busca em um funtor. Os exemplos acima discutem estratégias de busca em uma floresta fixa T. No nosso *topos*, porém, um objeto não é uma floresta, e sim um funtor G que mapeia problemas em florestas. Isto torna a definição de busca um pouco mais complexa: uma estratégia de busca, aqui, deve determinar uma maneira de percorrer todas as florestas na imagem de G "simultaneamente", de forma "natural".

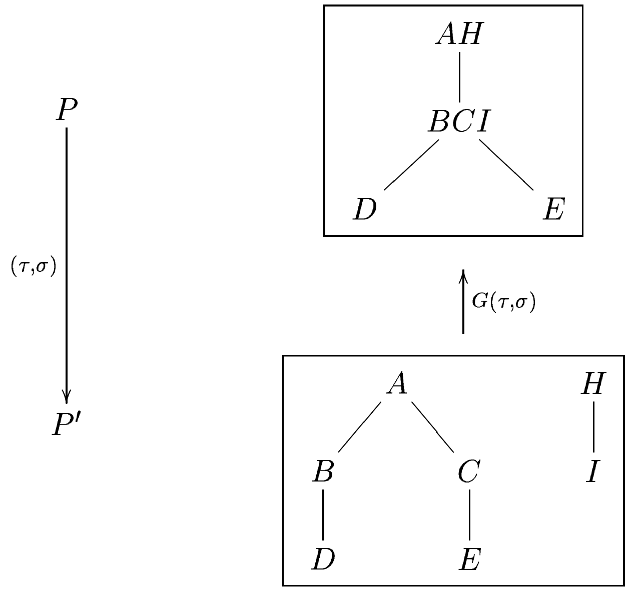
A idéia é manter a intuição de que uma estratégia de busca define uma sequência de sub-árvores de uma floresta, mas agora deveremos ter uma sequência de sub-árvores da floresta GP para cada problema P. No caso de uma redução $P \xrightarrow{(\tau,\sigma)} P'$, a sequência de sub-árvores de GP' deve estar relacionada à sequência de sub-árvores de GP através do homomorfismo $G(\tau,\sigma)$, como mostra o diagrama

$$
\begin{tikzcd}
 & & GP & & \\
P \arrow[ddd, "(\tau,\sigma)"'] & H_1P \arrow[ur] \arrow[rr, tail] & & H_2P \arrow[ul] \arrow[r, tail] & \cdots \\
 & & & & \\
 & & GP' \arrow[uuu, "G(\tau,\sigma)"] & & \\
P' & H_1P' \arrow[ur] \arrow[rr, tail] & & H_2P' \arrow[ul] \arrow[r, tail] & \cdots
\end{tikzcd}
$$

$\mathbf{Prob}_0$ **Floresta**

Nos próximos parágrafos, trabalharemos as definições para pôr em prática esta idéia. Mais adiante, expressaremos estas definições na linguagem de teoria local dos conjuntos.

5.2.2 Funtores colapsantes. Para manter a "naturalidade" do esquema descrito acima, gostaríamos que, dada uma redução $P \xrightarrow{(\tau,\sigma)} P'$, a escolha de um nó em GP correspondesse, sempre que possível, à escolha de um nó em GP'. Isto, no entanto, depende do homomorfismo $G(\tau, \sigma)$. Quando este homomorfismo não for injetivo sobre os nós, a escolha de um nó de GP pode corresponder à escolha de dois ou mais nós (talvez mesmo em árvores diferentes) de GP'. Na figura abaixo, por exemplo,



$\mathbf{Prob}_0$ **Floresta**

a escolha do nó BCI em GP corresponderia à escolha dos nós B, C e I em GP'. Mais adiante, veremos que uma estratégia de busca deve escolher, a cada passo, no máximo um nó da floresta associada a cada problema; por isso, a situação retratada na figura acima é indesejável.

5.2.3 Definição: funtor colapsante. Um funtor G como o mostrado acima, que mapeie alguma redução (τ, σ) em um homomorfismo $G(\tau, \sigma)$ não-injetivo sobre os nós, será chamado de funtor *colapsante* (por causa do fato de que dois ou mais nós de GP' são colapsados em um único nó de GP).

No restante desta seção, restringir-nos-emos apenas a funtores G não-colapsantes. Mais tarde, mostraremos como caracterizar esta classe de funtores em teoria local de conjuntos.

5.2.4 Cordas. Em 5.1.2, vimos que uma estratégia de busca em uma floresta visita um nó a cada iteração. Aqui, porém, em vez de uma floresta fixa, temos um objeto G de $\mathbf{ECF}$, que associa uma floresta GP a cada problema P. Definimos, então, a noção de *corda* de G, análoga à noção de nó de uma floresta. Uma corda de G é um conjunto especial de nós contendo no máximo um nó de cada floresta GP e satisfazendo certas condições:

5.2.5 Definição: corda. Seja G um funtor não-colapsante, objeto de $\mathbf{ECF}$; seja $G(\mathbf{Prob}_0)$ o conjunto de todas as florestas que G associa aos problemas em $\mathbf{Prob}_0$; seja N a união dos conjuntos de nós de todas estas florestas. Uma corda H de G é uma função parcial

$$H : G(\mathbf{Prob}_0) \to N$$

satisfazendo as seguintes condições:

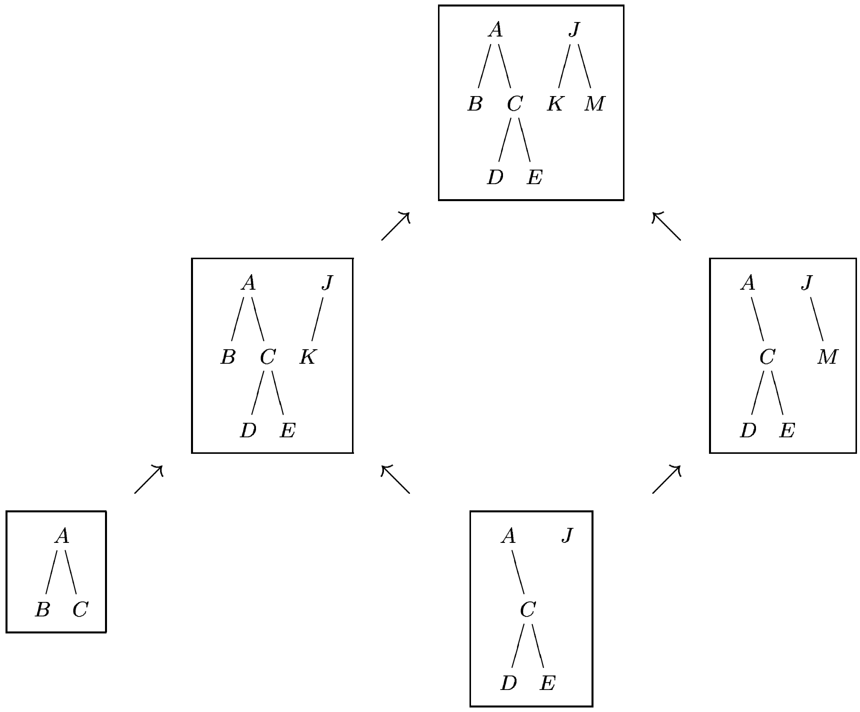
- H é definida para pelo menos um problema;
- Para todo problema P, se $H(P)$ for definida, então $H(P)$ é um nó de GP;
- Para toda redução $P \xrightarrow{(\tau,\sigma)} P'$ em $\mathbf{Prob}_0$ e para todo nó v' de GP':

$$(H(P) \text{ é definida e } G(\tau,\sigma)(v') \text{ é igual a, ou ancestral de, } H(P))$$
$$\Updownarrow$$
$$(H(P') \text{ é definida e } v' \text{ é igual a, ou ancestral de, } H(P'))$$

5.2.6 Exemplos de cordas. Mais uma vez, para facilitar a exposição, usaremos a seguinte categoria finita (a mesma do exemplo de 4.3) como $\mathbf{Prob}_0$.

$$\begin{tikzcd}
 & & P_1 \arrow[dl] \arrow[dr] & \\
 & P_2 \arrow[dl] \arrow[dr] & & P_4 \arrow[dl] \\
P_3 & & P_5 &
\end{tikzcd}$$

Suponha que G seja um funtor não-colapsante de $\mathbf{Prob}_0$ para $\mathbf{Floresta}$ que associa as seguintes florestas GP_1, GP_2, GP_3, GP_4 e GP_5 aos problemas acima. (Usamos a mesma notação do exemplo 4.4.2 para representar as florestas e os homomorfismos definidos por G.)



As seguintes funções parciais são cordas de G:

	P_1	P_2	P_3	P_4	P_5
H_1	A	A	A	A	A
H_2	C	C	C	C	C
H_3	B	B	B	A	A
H_4	J	J	$-$	J	J
H_5	K	K	$-$	J	J

Já as funções parciais abaixo não satisfazem a definição de corda:

	P_1	P_2	P_3	P_4	P_5
H_6	C	A	$-$	C	A
H_7	D	D	$-$	D	$-$
H_8	K	J	$-$	$-$	J
H_9	A	B	A	A	A
H_{10}	$-$	B	A	A	A

H_6 não é corda porque

– Na redução $P_1 \longrightarrow P_2$, temos que $H(P_1) = C$ e $H(P_2)$ é definida como A, mas o nó C de GP_2 (de quem $H(P_1)$ é imagem) não é igual a, nem ancestral de, A;

- Na redução $P_2 \longrightarrow P_3$, temos que $H(P_2) = A$, que é imagem do nó A de GP_3, mas $H(P_3)$ não é definida;
- Na redução $P_4 \longrightarrow P_5$, temos que $H(P_4) = C$ e $H(P_5)$ é definida como A, mas o nó C de GP_5 (de quem $H(P_4)$ é imagem) não é igual a, nem ancestral de, A.

H_7 não é corda porque

- Na redução $P_2 \longrightarrow P_3$, temos que $H(P_2) = D$; o nó C de GP_2 é ancestral de D é imagem do nó C de GP_3, mas $H(P_3)$ não é definida;
- Na redução $P_4 \longrightarrow P_5$, temos que $H(P_4) = D$; que é imagem do nó D de GP_5, mas $H(P_5)$ não é definida.

H_8 não é corda porque

- Na redução $P_1 \longrightarrow P_2$, temos que $H(P_1) = K$ e $H(P_2)$ é definida como J, mas o nó K de GP_2 (de quem $H(P_1)$ é imagem) não é igual a, nem ancestral de, J.

H_9 não é corda porque

- Na redução $P_1 \longrightarrow P_2$, temos que $H(P_2)$ é definida como B, mas sua imagem em GP_1 não é ancestral de, nem igual a, $H(P_1)$, que é A.

H_{10} não é corda porque

- Na redução $P_1 \longrightarrow P_2$, temos que $H(P_2)$ é definida como B, mas $H(P_1)$ não é definida.

Informalmente falando, os exemplos acima mostram que uma corda é um estágio em uma busca simultânea que caminha de forma “síncrona” em todas as florestas sempre que isto for possível.

5.3
Teoria Local dos Conjuntos

5.3.1 Apresentação. Esta seção explica como todo *topos* pode ser visto como um modelo de alguma teoria local de conjuntos e apresenta a linguagem usada no restante deste capítulo.

5.3.2 Tipos em vez de conjuntos. Em uma teoria local de conjuntos, o conceito de conjunto é substituído pelo de *tipo*. De fato, trata-se de uma linguagem tipada, onde cada entidade (inclusive conjuntos) habita um tipo específico. Daí também advém a idéia de *localidade*, pois as operações só estão definidas para entidades do mesmo tipo. Fora isso, a linguagem em muito se assemelha à linguagem de teoria de conjuntos, com os símbolos primitivos $=$, $\in$, $\{ \mid \}$.

5.3.3 Linguagem. A linguagem $\mathcal{L}$ de uma teoria local de conjuntos consiste nas seguintes classes:

- Os *símbolos* de $\mathcal{L}$ são o *símbolo do tipo unidade* $\mathbf{1}$, o *símbolo do tipo de valores-verdade* $\mathbf{\Omega}$, uma coleção de *símbolos básicos* $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$, e uma coleção de *símbolos de função* $\mathbf{f}, \mathbf{g}, \mathbf{h}, \ldots$;
- Os *tipos* de $\mathcal{L}$ são definidos como o menor conjunto $\mathcal{T}$ que contém $\mathbf{1}, \mathbf{\Omega}$, todos os símbolos básicos $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$ e que é fechado para as seguintes operações:
  - Para $\mathbf{A} \in \mathcal{T}$, o *tipo potência* $\mathbf{PA}$ também está em $\mathcal{T}$;
  - Para $\mathbf{A_1}, \ldots, \mathbf{A_n} \in \mathcal{T}$, o *tipo produto* $\mathbf{A_1} \times \cdots \times \mathbf{A_n}$ também está em $\mathcal{T}$ (para $n = 0$, o tipo produto é $\mathbf{1}$).
- Cada símbolo de função $\mathbf{f}$ está associado a uma *assinatura* $\mathbf{A} \to \mathbf{B}$, onde $\mathbf{A}$ e $\mathbf{B}$ são tipos. Isto é denotado por $\mathbf{f} : \mathbf{A} \to \mathbf{B}$;
- Para cada tipo $\mathbf{A}$ existe um conjunto enumerável de *variáveis* $V_{\mathbf{A}}$;
- Para cada tipo $\mathbf{A}$, existe um conjunto $T_{\mathbf{A}}$ de *termos de tipo* $\mathbf{A}$, definido como
  - $\star \in T_{\mathbf{1}}$;
  - $V_{\mathbf{A}} \subseteq T_{\mathbf{A}}$;
  - Para $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ e $\tau \in T_{\mathbf{A}}$, temos que $\mathbf{f}(\tau) \in T_{\mathbf{B}}$;

- Para $\tau_i \in T_{\mathbf{A}_i} (i = 1, \ldots, n)$, temos que $(\tau_1, \ldots, \tau_n) \in T_{\mathbf{A_1} \times \cdots \times \mathbf{A_n}}$. Para $n = 0$, este termo é $\star$;
- Para $\tau \in T_{\mathbf{A_1} \times \cdots \times \mathbf{A_n}}$, temos que $\pi_i(\tau) \in T_{\mathbf{A}_i}$ (com $i = 1, \ldots, n$);
- Para $\varphi \in T_{\mathbf{\Omega}}$ e $x \in V_{\mathbf{A}}$, temos que $\{x \mid \varphi\} \in T_{\mathbf{PA}}$;
- Para termos σ e τ do mesmo tipo $\mathbf{A}$, temos que $\sigma = \tau$ é um termo em $T_{\mathbf{\Omega}}$;
- Para termos σ e τ de tipos $\mathbf{A}$ e $\mathbf{PA}$, respectivamente, temos que $\sigma \in \tau$ é um termo em $T_{\mathbf{\Omega}}$.

Os termos de tipo $\mathbf{\Omega}$ são chamados de *fórmulas*.

5.3.4 Operações lógicas.

Os conectivos e quantificadores usuais são definidos como abreviaturas. No que se segue, $\varphi, \psi \in T_{\mathbf{\Omega}}$, e $\omega \in V_{\mathbf{\Omega}}$:

$$\begin{array}{lcl}
\varphi \Leftrightarrow \psi & \equiv & \varphi = \psi \\
\top & \equiv & \star = \star \\
\varphi \wedge \psi & \equiv & (\varphi, \psi) = (\top, \top) \\
\varphi \Rightarrow \psi & \equiv & (\varphi \wedge \psi) \Leftrightarrow \varphi \\
\forall x : \varphi & \equiv & \{x \mid \varphi\} = \{x \mid \top\} \\
\bot & \equiv & \forall \omega : \omega \\
\neg \varphi & \equiv & \varphi \Rightarrow \bot \\
\varphi \vee \psi & \equiv & \forall \omega : [(\varphi \Rightarrow \omega \wedge \psi \Rightarrow \omega) \Rightarrow \omega] \\
\exists x : \varphi & \equiv & \forall \omega : [\forall x : (\varphi \Rightarrow \omega) \Rightarrow \omega]
\end{array}$$

[13, pp. 71seg] apresenta um cálculo de sequentes para esta lógica (com as noções usuais de variáveis livres e ligadas, substituição de variáveis, etc.) e mostra que os conectivos e quantificadores definidos acima comportam-se como seus análogos na lógica intuicionista.

5.3.5 Conjuntos.

Em uma linguagem local $\mathcal{L}$, um *conjunto* é definido como um termo τ de tipo potência $\mathbf{PA}$ para algum tipo $\mathbf{A}$, tal que τ não possui variáveis livres[1]. Usamos as abreviaturas usuais (com os tipos sub-entendidos pelo contexto)

$$\begin{array}{lcl}
\forall x \in A : \varphi & \equiv & \forall x : (x \in A \Rightarrow \varphi) \\
\exists x \in A : \varphi & \equiv & \exists x : (x \in A \wedge \varphi) \\
\{x \in A \mid \varphi\} & \equiv & \{x \mid x \in A \wedge \varphi\} \\
\exists! x \in A : \varphi & \equiv & \exists x \in A : [\varphi \wedge \forall y \in A : (\varphi(x/y) \Rightarrow x = y)]
\end{array}$$

[1]Note que, no termo $\{x \mid \varphi\}$, a variável x encontra-se ligada, e não livre.

onde $\varphi(x/y)$ denota, como de costume, a fórmula φ com todas as ocorrências livres da variável x substituídas por y.

Podemos, então, definir operações de teoria dos conjuntos na lógica como, por exemplo (com X e Y conjuntos):

- $X \subseteq Y$ representa a fórmula $\forall x \in X : x \in Y$;
- $X \cap Y$ representa o conjunto $\{x \mid x \in X \land x \in Y\}$, do mesmo tipo que X e Y;
- $X \cup Y$ representa o conjunto $\{x \mid x \in X \lor x \in Y\}$, do mesmo tipo que X e Y;
- A representa o conjunto $\{x \mid \top\}$, do tipo $\mathbf{A}$, com x uma variável do tipo $\mathbf{A}$. Em outras palavras, para todo símbolo de tipo $\mathbf{A}$, existe um conjunto A correspondente. Ao considerarmos a interpretação da linguagem em um *topos*, vemos que isto significa que, para cada objeto A do *topos*, existe um conjunto A, de tipo $\mathbf{PA}$, na linguagem interna do *topos*;
- $\varnothing_{\mathbf{A}}$, ou simplesmente $\varnothing$, representa o conjunto $\{x \mid \bot\}$, de tipo $\mathbf{PA}$, com x uma variável do tipo $\mathbf{A}$;
- PX representa o conjunto $\{x \mid x \subseteq X\}$, de tipo $\mathbf{PPX}$, com x uma variável do tipo $\mathbf{PX}$;
- $\{\tau \mid \varphi\}$ representa o conjunto $\{x \mid \exists x_1 : \cdots : \exists x_n : (x = \tau \land \varphi)\}$, com x uma variável do mesmo tipo que o termo τ;
- $X \times Y$ representa o conjunto $\{(x, y) \mid x \in X \land y \in Y\}$, de tipo $\mathbf{P}(\mathbf{X} \times \mathbf{Y})$. Note que X e Y podem ser de tipos diferentes;
- $X + Y$ representa o conjunto $\{(\{x\}, \varnothing) \mid x \in X\} \cup \{(\varnothing, \{y\}) \mid y \in Y\}$, de tipo $\mathbf{P}(\mathbf{PX} \times \mathbf{PY})$. Note que X e Y podem ser de tipos diferentes;
- Y^X representa o conjunto $\{z \mid z \subseteq X \times Y \land \forall x \in X : \exists! y \in Y : (x, y) \in z\}$, de tipo $\mathbf{PP}(\mathbf{X} \times \mathbf{Y})$. [13, pp. 85seg] mostra que a cada símbolo de função $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ da linguagem, corresponde o conjunto $\{(a, \mathbf{f}(a)) \mid a \in A\}$, de tipo B^A. Este fato, quando interpretado em um *topos*, significa que a cada morfismo $A \xrightarrow{f} B$ está associado um conjunto f de tipo B^A. Em outras palavras, podemos representar morfismos do mesmo modo que funções em teoria de conjuntos clássica: como conjuntos de pares;
- $\coprod_{i \in I} X_i$ representa o conjunto $\{(i, x) \mid i \in I \land x \in X_i\}$ de tipo $\mathbf{P}(\mathbf{B} \times \mathbf{A})$, com I de tipo $\mathbf{B}$ e com X_i um termo de tipo $\mathbf{PA}$ contendo ou não ocorrências livres da variável i.

5.3.6 Interpretação. [13, pp. 91seg] explica em detalhes como uma linguagem local pode ser interpretada em um *topos* arbitrário. Basicamente, uma interpretação I da linguagem $\mathcal{L}$ em um *topos* $\mathbf{E}$ consiste em

- Uma atribuição, a cada tipo $\mathbf{A}$, de um objeto A de $\mathbf{E}$ de maneira que
  - Todo tipo $\mathbf{A_1} \times \cdots \times \mathbf{A_n}$ é associado ao objeto $A_1 \times \cdots \times A_n$;
  - Todo tipo $\mathbf{PA}$ é associado ao objeto PA;
  - O tipo $\mathbf{1}$ é associado ao objeto inicial 1;
  - O tipo $\mathbf{\Omega}$ é associado ao objeto de valores-verdade Ω.
- Uma atribuição, a cada símbolo de função $\mathbf{f} : \mathbf{A} \to \mathbf{B}$, de um morfismo $A \xrightarrow{f} B$.

[13, p. 92] mostra como estender uma interpretação a todos os termos da linguagem $\mathcal{L}$. Basicamente, dados um termo τ de tipo $\mathbf{B}$ e uma lista de variáveis $\vec{x} = (x_1, \ldots, x_n)$ que contenha todas as variáveis livres de τ (e possivelmente outras variáveis), a interpretação de τ com as variáveis $\vec{x}$ é um morfismo

$$A_1 \times \cdots \times A_n \xrightarrow{[\![\tau]\!]_{\vec{x}}} B$$

onde $\mathbf{A_1}, \ldots, \mathbf{A_n}$ são os tipos das variáveis em $\vec{x}$. Em especial, se τ for um termo fechado (sem variáveis livres) do tipo $\mathbf{B}$, a interpretação de τ com $\vec{x} = \varnothing$ é um morfismo

$$1 \xrightarrow{[\![\tau]\!]_{\varnothing}} B$$

ou seja, um elemento global de B.

No caso particular de uma fórmula φ, a interpretação de φ com a lista de variáveis $\vec{x}$ é um morfismo

$$A_1 \times \cdots \times A_n \xrightarrow{[\![\varphi]\!]_{\vec{x}}} \Omega$$

onde $\mathbf{A_1}, \ldots, \mathbf{A_n}$ são os tipos das variáveis em $\vec{x}$, e Ω é o objeto de valores-verdade do *topos*.

Se φ é uma sentença (uma fórmula sem variáveis livres), a interpretação de φ com $\vec{x} = \varnothing$ é um morfismo

$$1 \xrightarrow{[\![\varphi]\!]_{\varnothing}} \Omega$$

isto é, um valor-verdade de Ω.

5.3.7 Sequentes. Um sequente em uma linguagem local $\mathcal{L}$ é um par (Γ, φ), onde Γ é um conjunto finito de fórmulas e φ é uma fórmula. Um sequente é escrito na forma

$$\Gamma : \varphi$$

Definimos a interpretação de $\Gamma = \{\psi_1, \ldots, \psi_m\}$ não-vazio com a lista de variáveis $\vec{x}$ como

$$[\![\Gamma]\!]_{\vec{x}} = [\![\psi_1 \wedge \cdots \wedge \psi_m]\!]_{\vec{x}}$$

Para $\Gamma = \varnothing$, definimos

$$[\![\Gamma]\!]_{\vec{x}} = [\![\top]\!]_{\vec{x}}$$

Seja $\vec{x} = (x_1, \ldots, x_n)$ uma lista contendo todas as variáveis livres em $\Gamma \cup \{\varphi\}$. Então, tanto $[\![\Gamma]\!]_{\vec{x}}$ quanto $[\![\varphi]\!]_{\vec{x}}$ são morfismos do objeto $A_1 \times \cdots \times A_n$ para Ω, onde $\mathbf{A_1}, \ldots, \mathbf{A_n}$ são os tipos das variáveis em $\vec{x}$.

Pelo funcionamento do classificador de sub-objetos, estes dois morfismos podem ser vistos como os morfismos característicos de dois sub-objetos de $A_1 \times \cdots \times A_n$. Chamemos estes dois sub-objetos de S_Γ e de S_φ, respectivamente. Dizemos, então, que um sequente $\Gamma : \varphi$ é válido na interpretação I, escrito

$$\Gamma \models_I \varphi$$

se e somente se ocorrer que

$$S_\Gamma \subseteq S_\varphi$$

onde "$\subseteq$" é a relação de continência entre sub-objetos de $A_1 \times \cdots \times A_n$.

5.3.8 A lógica interna de um *topos*. Agora podemos definir com precisão o que é a lógica interna $Th(\mathbf{E})$ de um *topos* $\mathbf{E}$. A linguagem $\mathcal{L}(\mathbf{E})$ desta lógica consiste de um símbolo $\mathbf{A}$ para cada objeto $\mathbf{A}_E$ de $\mathbf{E}$ que não 1 ou Ω, de tal maneira que

$$\begin{aligned} \mathbf{A}_E &= A \text{ para } \mathbf{A} \text{ básico} \\ (\mathbf{A} \times \mathbf{B})_E &= \mathbf{A}_E \times \mathbf{B}_E \\ (\mathbf{PA})_E &= P(\mathbf{A}_E) \end{aligned}$$

e de um símbolo de função $(\mathbf{f}, \mathbf{A}, \mathbf{B})$ para cada morfismo $A \xrightarrow{f} B$ em $\mathbf{E}$.

A *interpretação natural* da linguagem $\mathcal{L}(\mathbf{E})$ no *topos* $\mathbf{E}$ é a que associa o símbolo básico $\mathbf{A}_E$ ao objeto A e o símbolo de função $(\mathbf{f}, \mathbf{A}, \mathbf{B})$ ao morfismo $A \xrightarrow{f} B$.

A *lógica interna* $Th(\mathbf{E})$ de $\mathbf{E}$, também chamada de a *teoria* de $\mathbf{E}$, é a teoria, na linguagem $\mathcal{L}(\mathbf{E})$, cujos axiomas são todos os sequentes $\Gamma : \varphi$ válidos na interpretação natural.

5.3.9 Notação. Antes de examinar exemplos, algumas observações sobre a notação utilizada no restante do trabalho. Dada a linguagem local $\mathcal{L}(\mathbf{E})$ de um *topos* $\mathbf{E}$, temos que:

- Os símbolos (básicos, de função, etc.) são escritos em negrito (e.g., $\mathbf{A}$, $\mathbf{f}$, $\mathbf{A} \times \mathbf{B}$, $\mathbf{PA}$);
- Os termos que não forem representados por símbolos não-alfabéticos (como $\star$, $\top$) são escritos em itálico (e.g. x, (x, y));
- Os tipos são escritos em negrito (e.g. $\mathbf{A}$), mas como a cada tipo $\mathbf{A}$ corresponde um termo A de tipo $\mathbf{PA}$, muitas vezes representaremos o tipo pelo termo correspondente, escrevendo seu nome em itálico;
- O mesmo ocorre com símbolos de função: como cada símbolo de função $(\mathbf{f}, \mathbf{A}, \mathbf{B})$ corresponde a um termo f de tipo $\mathbf{B}^{\mathbf{A}}$, escrevemos simplesmente f;
- Predicados definidos na linguagem são grafados em negrito;
- Na próxima seção, onde o *topos* de interesse é $\mathbf{ECF}$, termos como $\langle\, H, \kappa \,\rangle$ são usados como variáveis. Isto pode ser entendido como uma abreviatura através da qual, por exemplo, a fórmula

  $$\forall \langle\, H, \kappa \,\rangle : \varphi$$

  representa

  $$\forall x : \forall H : \forall \kappa : [(\pi_1(x) = H \ \wedge \ \pi_2(x) = \kappa) \Rightarrow \varphi]$$

  com os tipos sub-entendidos no contexto.

5.3.10 Exemplos.

- Dados dois objetos A e B, o predicado que afirma que um morfismo $A \xrightarrow{f} B$ é um monomorfismo é

$$\boxed{\begin{array}{l} \mathbf{isMonic}(f, A, B) \Leftrightarrow \\ \quad \forall x \in A : \forall y \in A : (f(x) = f(y) \Rightarrow x = y) \end{array}}$$

– Dados dois objetos A e B, o predicado que afirma que um morfismo $A \xrightarrow{f} B$ é um epimorfismo é

$$\boxed{\begin{array}{l}\mathbf{isEpic}(f, A, B) \Leftrightarrow \\ \quad \forall y \in B : \exists x \in A : f(x) = y\end{array}}$$

– Em um *topos*, todo morfismo $A \xrightarrow{f} B$ pode ser decomposto, de forma única, em um monomorfismo g e um epimorfismo h tais que $g \circ h = f$. O domínio de g (e codomínio de h) é o sub-objeto de B dado pelo termo

$$\{y \in B \mid \exists x \in A : f(x) = y\}$$

Chamamos este termo de $image(f)$.

– Dado um monomorfismo $A \overset{m}{\rightarrowtail} B$, o morfismo característico $B \xrightarrow{\chi_m} \Omega$ é representado pelo seguinte termo, de tipo $\mathbf{\Omega^B}$ (um subtipo de $\mathbf{PP(B \times \Omega)}$):

$$\{(b, \omega) \in P(B \times \Omega) \mid \omega = (\exists x \in A : b = m(x))\}$$

Chamaremos este termo de χ_m, ou de χ_A, conforme a ênfase seja no morfismo m ou no domínio A de m.

5.4
Predicados e Termos Úteis

5.4.1 Apresentação. Nesta seção, definiremos predicados e termos na linguagem interna de **ECF** que serão utilizados na especificação de estratégias de busca. Basicamente, traduziremos a noção de corda, definida em 5.2, para a linguagem interna de **ECF**. Em seguida, definiremos predicados e termos relativos a raízes, árvores, folhas etc. Para iniciar, porém, algumas observações sobre os rótulos dos nós das florestas.

5.4.2 Separando forma e rotulamento. Conforme discutido em 4.1, um objeto G do *topos* **ECF** não possui informação sobre o rotulamento das florestas associadas aos problemas. Esta informação adicional é dada por um morfismo (em **ECF**) $G \xrightarrow{\lambda} L$, onde L é o objeto definido em 3.4.4. Assim, os predicados e termos que definiremos a seguir envolverão termos G e λ, onde G é responsável pela forma das florestas, e λ é responsável pelo rotulamento das florestas. Um benefício adicional desta abordagem é a facilidade de definir predicados e termos que dizem respeito apenas à forma, ou apenas ao rotulamento, das florestas associadas aos problemas. No primeiro caso, apenas G é utilizado; no segundo, apenas λ.

5.4.3 Morfismos que preservam o rotulamento. Em 3.4.6, vimos que um morfismo $\langle\, G, \lambda \,\rangle \xrightarrow{\alpha} \langle\, G', \lambda' \,\rangle$ em **ECFR** é uma transformação natural $\alpha : G \dot{\to} G'$ tal que α preserva o rotulamento das florestas escolhidas por G; i.e., $\lambda' \circ \alpha = \lambda$. O seguinte predicado serve para verificar se um morfismo $G \xrightarrow{\alpha} G'$ em **ECF** possui esta propriedade:

$$\boxed{\begin{array}{l} \mathbf{isLabelPreserving}(\alpha, G, \lambda, G', \lambda') \Leftrightarrow \\ \quad \forall x \in G : \lambda(x) = \lambda'(\alpha(x)) \end{array}}$$

com α do tipo $\mathbf{G'^{G}}$, λ do tipo $\mathbf{L^{G}}$ e λ' do tipo $\mathbf{L^{G'}}$.

5.4.4 Sub-objetos em ECFR. Dados dois objetos de **ECFR** $\langle\, H, \kappa \,\rangle$ e $\langle\, G, \lambda \,\rangle$, com $H \overset{h}{\rightarrowtail} G$ um sub-objeto em **ECF**, temos que $\langle\, H, \kappa \,\rangle$ será sub-objeto de $\langle\, G, \lambda \,\rangle$ em **ECFR** se e somente se h preservar o rotulamento dado por κ. Isto é expresso pelo seguinte predicado:

$$\boxed{\begin{array}{l} \mathbf{isSubObject}(H, \kappa, G, \lambda) \Leftrightarrow \\ \quad \exists h \in G^H : (\\ \qquad \mathbf{isMonic}(h) \ \wedge \ \mathbf{isLabelPreserving}(h, H, \kappa, G, \lambda) \\ \quad) \end{array}}$$

com κ do tipo $\mathbf{L^{H}}$, λ do tipo $\mathbf{L^{G}}$, e **isMonic** o predicado que afirma que um morfismo é um monomorfismo.

5.4.5 Nós de uma floresta = caminhos finitos e não-vazios. Cada nó de uma floresta determina e é determinado por um caminho finito e não-vazio iniciando-se em alguma raiz da floresta. Isto nos leva à seguinte proposição:

5.4.6 Proposição: toda corda H de um objeto G de ECF é sub-objeto de G. Mais especificamente, o sub-objeto H é um funtor que associa a cada problema P uma *árvore* linear e finita HP, de tal forma que, para algum P, a árvore HP é não-vazia.

5.4.7 Caracterizando sub-objetos que são cordas. Nem todos os sub-objetos H de um objeto G de **ECF** correspondem a cordas, porém. Além da exigência de que HP seja uma árvore, linear e finita para cada problema P, devemos exigir que H satisfaça a noção de "sincronicidade" informalmente apresentada em 5.2.6. Por exemplo, as funções parciais H_6 e H_7 de 5.2.6 definem sub-objetos de G, mas não satisfazem a definição de corda.

Para caracterizar os sub-objetos $H \overset{h}{\rightarrowtail} G$ que são cordas, imporemos uma condição sobre o morfismo característico $G \xrightarrow{\chi_H} \Omega$. Lembre-se, de 5.2.3, que estaremos considerando apenas funtores não-colapsantes:

5.4.8 Proposição: seja G um funtor não-colapsante, objeto de **ECF**; seja H um sub-objeto de G tal que, para todo problema P, ocorre que HP é uma árvore linear finita e tal que, para algum problema P, ocorre que HP é não-vazia; então, $H \overset{h}{\rightarrowtail} G$ é uma corda se e somente se, para todo problema P, ocorrer que $GP \xrightarrow{\chi_{HP}} \Omega(P)$ é um homomorfismo de florestas cuja imagem em $\Omega(P)$ inclui apenas nós representados (usando a notação de 4.2.9) por tuplas da forma

$$(n, n, \ldots)$$

ou

$$(-, -, \ldots)$$

com todas as componentes iguais.

5.4.9 Um predicado para caracterizar cordas. Representemos[2] por Ω_c o sub-objeto de Ω tal que, para todo problema P, ocorre que $\Omega_c(P)$ é a subfloresta de $\Omega(P)$ que consiste dos nós representados por tuplas com todas as componentes iguais. Em vista do discutido acima, podemos definir o predicado **isString** na linguagem interna

[2]Como estamos trabalhando na teoria $Th(\mathbf{ECF})$, que inclui um símbolo para cada objeto de **ECF**, não nos preocuparemos em definir Ω_C sintaticamente (assim como L também não foi definido sintaticamente).

de **ECF** para expressar o fato de que um sub-objeto $\langle\, H, \kappa \,\rangle$ de $\langle\, G, \lambda \,\rangle$ é uma corda:

$$\boxed{\begin{array}{l} \mathbf{isString}(H, \kappa, G, \lambda) \Leftrightarrow \\ \quad \mathbf{isSubObject}(H, \kappa, G, \lambda) \wedge \\ \quad H \neq \varnothing_G \wedge \\ \quad \mathbf{isMonic}(1_H) \wedge \\ \quad \neg\mathbf{isEpic}(1_H) \wedge \\ \quad \mathit{image}(\chi_H) \subseteq \Omega_c \end{array}}$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$.

As seguintes observações se aplicam:

- $\varnothing_G$ representa o objeto inicial, visto como um objeto do tipo $\mathbf{G}$;
- 1_H representa o único morfismo de H para o objeto terminal 1;
- **isMonic** e **isEpic** são predicados que significam que um morfismo é um monomorfismo e um epimorfismo, respectivamente.
- $\mathit{image}(\chi_H)$ representa a imagem em Ω do morfismo característico de H;
- Na conjunção acima, a primeira fórmula garante que temos um sub-objeto de $\langle\, G, \lambda \,\rangle$;
- A segunda fórmula diz que H não é o sub-objeto inicial (i.e., para algum P, a floresta HP é não-vazia);
- O fato de 1_H ser um monomorfismo equivale a HP ser uma árvore linear para todo problema P;
- O fato de 1_H não ser um epimorfismo significa que HP é uma árvore finita para todo problema P.

5.4.10 O objeto de cordas de $\langle\, G, \lambda \,\rangle$. O seguinte termo denota o objeto de cordas de um objeto $\langle\, G, \lambda \,\rangle$ de **ECFR**:

$$\boxed{\left\{ \langle\, H, \kappa \,\rangle \in \coprod_{H \in PG} L^H \;\middle|\; \mathbf{isString}(H, \kappa, G, \lambda) \right\}}$$

onde

$$\coprod_{H \in PG} L^H = \{\langle\, H, \kappa \,\rangle \mid H \in PG \wedge \kappa \in L^H\}$$

Usaremos a abreviatura $\mathit{strings}(G, \lambda)$ para este termo, com λ do tipo $\mathbf{L^G}$.

5.4.11 Ordem entre cordas. A ordem parcial "$\subseteq$" entre sub-objetos de um objeto $\langle\, G, \lambda \,\rangle$ de $\mathbf{ECFR}$ é herdada pelas cordas de $\langle\, G, \lambda \,\rangle$. Assim, quando $\langle\, H, \kappa \,\rangle$ e $\langle\, H', \kappa' \,\rangle$ forem cordas de $\langle\, G, \lambda \,\rangle$, a fórmula

$$\boxed{\mathbf{isSubObject}(H, \kappa, H', \kappa')}$$

é abreviada por

$$\boxed{\langle\, H, \kappa \,\rangle \ \subseteq \ \langle\, H', \kappa' \,\rangle}$$

com κ do tipo $\mathbf{L^H}$, e κ' do tipo $\mathbf{L^{H'}}$.

5.4.12 Funtores não-colapsantes. Agora temos condições de verificar quando um objeto G de $\mathbf{ECF}$ é um funtor não-colapsante (5.2.3).

Considere a coleção de todos os sub-objetos finitos e lineares de G. Então, G é não-colapsante se e somente se todos os elementos maximais desta coleção (em relação a "$\subseteq$") forem cordas (i.e., se estes elementos satisfizerem a proposição 5.4.8). Isto é expresso pelo predicado:

$$\boxed{\begin{array}{l}
\mathbf{isNonCollapsing}(G) \Leftrightarrow \\
\quad \forall H \in PG : (\\
\qquad (H \neq 0_G \ \wedge \ \mathbf{isMonic}(1_H) \ \wedge \ \neg\mathbf{isEpic}(1_H) \ \wedge \\
\qquad\quad \neg\exists H' \in PG : (\\
\qquad\quad H' \neq 0_G \ \wedge \ \mathbf{isMonic}(1'_H) \ \wedge \ \neg\mathbf{isEpic}(1'_H) \ \wedge \ H \subseteq H' \\
\qquad\quad) \\
\qquad) \Rightarrow \\
\qquad \mathit{image}(\chi_H) \subseteq \Omega_c \\
\quad)
\end{array}}$$

5.4.13 Proposição: G é não-colapsante se e somente se

$$\mathbf{isNonCollapsing}(G)$$

é uma fórmula válida em $\mathbf{ECF}$.

5.4.14 Raízes. Uma raiz de um objeto $\langle\, G, \lambda \,\rangle$ de $\mathbf{ECFR}$ é um elemento minimal no conjunto parcialmente ordenado de cordas de $\langle\, G, \lambda \,\rangle$, com a ordem "$\subseteq$" definida acima:

$$\boxed{\begin{array}{l}\mathbf{isRoot}(H,\kappa,G,\lambda) \Leftrightarrow \\ \quad \langle\ H,\kappa\ \rangle\ \in \mathit{strings}(G,\lambda) \wedge \\ \quad \neg\exists\langle\ H',\kappa'\ \rangle\ \in \mathit{strings}(G,\lambda) :\ \langle\ H',\kappa'\ \rangle\ \subset\ \langle\ H,\kappa\ \rangle\end{array}}$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$, e onde, como de costume, $\langle\ H',\kappa'\ \rangle\ \subset\ \langle\ H,\kappa\ \rangle$ é abreviatura de

$$\langle\ H',\kappa'\ \rangle\ \subseteq\ \langle\ H,\kappa\ \rangle\ \wedge\ \langle\ H',\kappa'\ \rangle\ \neq\ \langle\ H,\kappa\ \rangle$$

5.4.15 Folhas. Uma folha de um objeto $\langle\ G,\lambda\ \rangle$ de **ECFR** é um elemento maximal no conjunto parcialmente ordenado de cordas de $\langle\ G,\lambda\ \rangle$, com a ordem "$\subseteq$":

$$\boxed{\begin{array}{l}\mathbf{isLeaf}(H,\kappa,G,\lambda) \Leftrightarrow \\ \quad \langle\ H,\kappa\ \rangle\ \in \mathit{strings}(G,\lambda) \wedge \\ \quad \neg\exists\langle\ H',\kappa'\ \rangle\ \in \mathit{strings}(G,\lambda) :\ \langle\ H,\kappa\ \rangle\ \subset\ \langle\ H',\kappa'\ \rangle\end{array}}$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$.

5.4.16 Árvores. Um objeto $\langle\ G,\lambda\ \rangle$ de **ECFR** é uma árvore não-vazia quando possui exatamente uma corda raiz:

$$\boxed{\begin{array}{l}\mathbf{isTree}(G,\lambda) \Leftrightarrow \\ \quad \exists!\langle\ H,\kappa\ \rangle\ \in \mathit{strings}(G,\lambda) : \mathbf{isRoot}(H,\kappa)\end{array}}$$

com λ do tipo $\mathbf{L^G}$.

5.4.17 O objeto de árvores de $\langle\ G,\lambda\ \rangle$. O seguinte termo denota o objeto de árvores não-vazias de um objeto $\langle\ G,\lambda\ \rangle$ de **ECFR**:

$$\boxed{\left\{\ \langle\ H,\kappa\ \rangle\ \in \coprod_{H\in PG} L^H\ \middle|\ \mathbf{isTree}(H,\kappa)\ \right\}}$$

onde

$$\coprod_{H\in PG} L^H\ =\ \{\langle\ H,\kappa\ \rangle\ \mid\ H\in PG\ \wedge\ \kappa\in L^H\}$$

Usaremos a abreviatura $\mathit{trees}(G,\lambda)$ para este termo, com λ do tipo $\mathbf{L^G}$.

5.4.18 Extensões. O fato de um objeto $\langle\ G, \lambda\ \rangle$ de **ECFR** ser uma extensão de um objeto $\langle\ G', \lambda'\ \rangle$ é representado por $\langle\ G', \lambda'\ \rangle\ \subset\ \langle\ G, \lambda\ \rangle$. Mais adiante, será conveniente dispor de um predicado para dizer que $\langle\ G, \lambda\ \rangle$ é uma extensão de $\langle\ G', \lambda'\ \rangle$ que possui exatamente uma corda a mais que $\langle\ G', \lambda'\ \rangle$:

$$\boxed{\begin{array}{l} \mathbf{extendsByOne}(G, \lambda, G', \lambda') \Leftrightarrow \\ \quad \langle\ G', \lambda'\ \rangle\ \subset\ \langle\ G, \lambda\ \rangle \wedge \\ \quad \exists! \langle\ H, \kappa\ \rangle\ \in \mathit{strings}(G, \lambda) :\ \langle\ H, \kappa\ \rangle\ \notin \mathit{strings}(G', \lambda') \end{array}}$$

com λ do tipo $\mathbf{L^G}$, λ' do tipo $\mathbf{L^{G'}}$.

5.4.19 Identificando a corda acrescentada. Se um objeto $\langle\ G, \lambda\ \rangle$ estende um objeto $\langle\ G', \lambda'\ \rangle$ por exatamente uma corda, o seguinte predicado diz que a corda acrescentada é $\langle\ H, \kappa\ \rangle$:

$$\boxed{\begin{array}{l} \mathbf{wasAdded}(H, \kappa, G, \lambda, G', \lambda') \Leftrightarrow \\ \quad \mathbf{extendsByOne}(G, \lambda, G', \lambda') \wedge \\ \quad \langle\ H, \kappa\ \rangle\ \in \mathit{strings}(G, \lambda)\ \wedge\ \langle\ H, \kappa\ \rangle\ \notin \mathit{strings}(G', \lambda') \end{array}}$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$, λ' do tipo $\mathbf{L^{G'}}$.

5.4.20 Sucessoras de uma corda. O predicado **extendsByOne**, quando aplicado a duas cordas $\langle\ H, \kappa\ \rangle$ e $\langle\ H', \kappa'\ \rangle$ de um objeto $\langle\ G, \lambda\ \rangle$, significa que $\langle\ H, \kappa\ \rangle$ é uma sucessora (filha) de $\langle\ H', \kappa'\ \rangle$:

$$\boxed{\begin{array}{l} \mathbf{isChild}(H, \kappa, H', \kappa', G, \lambda) \Leftrightarrow \\ \quad \langle\ H, \kappa\ \rangle\ \in \mathit{strings}(G, \lambda) \wedge \\ \quad \langle\ H', \kappa'\ \rangle\ \in \mathit{strings}(G, \lambda) \wedge \\ \quad \mathbf{extendsByOne}(H, \kappa, H', \kappa') \end{array}}$$

com κ do tipo $\mathbf{L^H}$, κ' do tipo $\mathbf{L^{H'}}$, λ do tipo $\mathbf{L^G}$.

5.4.21 O objeto de sucessoras de $\langle\ H, \kappa\ \rangle$. O seguinte termo denota o objeto de sucessoras de uma corda $\langle\ H, \kappa\ \rangle$ de um objeto $\langle\ G, \lambda\ \rangle$ de **ECFR**:

$$\boxed{\{\ \langle\ H', \kappa'\ \rangle\ \in \mathit{strings}(G, \lambda) \mid \mathbf{isChild}(H', \kappa', H, \kappa, G, \lambda)\}}$$

Usaremos a abreviatura $children(H, \kappa, G, \lambda)$ para este termo.

5.4.22 Descendentes de uma corda. Uma corda $\langle\, H, \kappa \,\rangle$ é descendente de uma corda $\langle\, H', \kappa' \,\rangle$ em um objeto $\langle\, G, \lambda \,\rangle$ se $\langle\, H', \kappa' \,\rangle$ for um sub- objeto próprio de $\langle\, H, \kappa \,\rangle$:

$$\boxed{\begin{array}{l} \mathbf{isDescendant}(H, \kappa, H', \kappa', G, \lambda) \Leftrightarrow \\ \quad \langle\, H, \kappa \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \langle\, H', \kappa' \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \langle\, H', \kappa' \,\rangle \ \subset \ \langle\, H, \kappa \,\rangle \end{array}}$$

com κ do tipo $\mathbf{L^H}$, κ' do tipo $\mathbf{L^{H'}}$, λ do tipo $\mathbf{L^G}$.

5.4.23 Antecessoras de uma corda. O predicado que diz que uma corda $\langle\, H, \kappa \,\rangle$ é antecessora de uma corda $\langle\, H', \kappa' \,\rangle$ em um objeto $\langle\, G, \lambda \,\rangle$ é simétrico ao predicado **isChild**:

$$\boxed{\begin{array}{l} \mathbf{isParent}(H, \kappa, H', \kappa', G, \lambda) \Leftrightarrow \\ \quad \langle\, H, \kappa \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \langle\, H', \kappa' \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \mathbf{extendsByOne}(H', \kappa', H, \kappa) \end{array}}$$

com κ do tipo $\mathbf{L^H}$, κ' do tipo $\mathbf{L^{H'}}$, λ do tipo $\mathbf{L^G}$.

5.4.24 Ancestrais de uma corda. Uma corda $\langle\, H, \kappa \,\rangle$ é ancestral de uma corda $\langle\, H', \kappa' \,\rangle$ em um objeto $\langle\, G, \lambda \,\rangle$ se $\langle\, H, \kappa \,\rangle$ for um sub-objeto próprio de $\langle\, H', \kappa' \,\rangle$:

$$\boxed{\begin{array}{l} \mathbf{isAncestor}(H, \kappa, H', \kappa', G, \lambda) \Leftrightarrow \\ \quad \langle\, H, \kappa \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \langle\, H', \kappa' \,\rangle \ \in strings(G, \lambda) \wedge \\ \quad \langle\, H, \kappa \,\rangle \ \subset \ \langle\, H', \kappa' \,\rangle \end{array}}$$

com κ do tipo $\mathbf{L^H}$, κ' do tipo $\mathbf{L^{H'}}$, λ do tipo $\mathbf{L^G}$.

5.4.25 Cordas exploradas. Mais adiante, precisaremos de um predicado que diz se uma corda $\langle\, H, \kappa \,\rangle$ de um objeto $\langle\, G, \lambda \,\rangle$ já foi explorada; i.e., se todas as cordas sucessoras de $\langle\, H, \kappa \,\rangle$ já foram visitadas. O predicado recebe a sub-árvore $\langle\, T, \theta \,\rangle$, que consiste das cordas de $\langle\, G, \lambda \,\rangle$ que já foram visitadas:

$$
\begin{array}{|l|}
\hline
\textbf{isExplored}(H, \kappa, G, \lambda, T, \theta) \Leftrightarrow \\
\quad \forall \langle\, H', \kappa' \,\rangle \in \mathit{strings}(\langle\, G, \lambda \,\rangle) : (\\
\quad\quad \textbf{isChild}(H', \kappa', H, \kappa, G, \lambda) \Rightarrow \langle\, H', \kappa' \,\rangle \in \mathit{strings}(\langle\, T, \theta \,\rangle) \\
\quad) \\
\hline
\end{array}
$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$, θ do tipo $\mathbf{L^T}$.

5.4.26 Um termo para a cardinalidade de um objeto finito. Dado um objeto A de **ECF**, definimos um termo para representar o menor natural n tal que existe um epimorfismo do sub-objeto $\{x \in N \mid x < n\}$ para A. Chamaremos este natural n de "cardinalidade" de A, embora isto só faça sentido quando A possuir uma quantidade finita de elementos globais e for extensional (no sentido de [9, p. 169]). Para nossos propósitos no capítulo 6, porém, esta definição será adequada. Assim,

$$
\begin{array}{|l|}
\hline
\mathit{card}(A) = \\
\quad \iota n \in N : (\\
\quad\quad ((\neg\exists f \in A^N : \textbf{isEpic}(f, N, A)) \;\wedge\; n = 0) \\
\quad\quad \vee \\
\quad\quad (\\
\quad\quad\quad \exists f \in A^{\{x \in N | x < n\}} : \textbf{isEpic}(f, \{x \in N \mid x < n\}, A) \\
\quad\quad\quad \wedge \\
\quad\quad\quad \forall m \in N : (\\
\quad\quad\quad\quad m < n \Rightarrow \\
\quad\quad\quad\quad\quad \neg\exists g \in A^{\{x \in N | x < m\}} : \textbf{isEpic}(g, \{x \in N \mid x < m\}, A) \\
\quad\quad\quad) \\
\quad\quad) \\
\quad) \\
\hline
\end{array}
$$

onde $\iota n \in N$ significa "o único $n \in N$".

5.4.27 Um termo para o somatório. Vimos, no capítulo 4, que o *topos* **ECF** possui um objeto de números naturais, o que permite a definição de termos por recursão simples ou por recursão primitiva sobre os naturais. Dado um objeto finito A, com cardinalidade $\mathit{card}(A)$, e um morfismo $A \xrightarrow{f} B$, onde B é o objeto dos naturais N ou o objeto dos racionais Q, queremos definir um termo para a soma

$$\sum_{a \in A} f(a)$$

Usamos recursão primitiva, então, para definir o termo $sum(A, f, n)$, do tipo **N** ou do tipo **Q**, conforme o codomínio do morfismo f, do seguinte modo:

$$\begin{aligned} sum(A, f, 0) &= 0 \\ sum(A, f, sn) &= \min\{sum(A - \{a\}, f, n) + f(a) \mid a \in A\} \end{aligned}$$

No capítulo 6, usaremos a notação usual

$$\sum_{a \in A} f(a)$$

para denotar o termo

$$sum(A, f, card(A))$$

5.4.28 Termos que representam problemas. Existem, em **ECF**, objetos que podem ser usados para representar os domínios de instâncias D e de respostas R, bem como as relações p, dos problemas em $\mathbf{Prob}_0$. Estes objetos serão úteis, mais adiante, na definição de heurísticas.

- Para representar os domínios de instâncias, o funtor

  $$G_D : \mathbf{Prob}_0 \to \mathbf{Floresta}$$

  associa cada problema $P = \langle\, D, R, p\,\rangle$ a uma floresta cujo conjunto de nós em cada nível i é exatamente o conjunto $\wp(D)$ de todos os subconjuntos de D, e cada nó tem como único filho a cópia de si mesmo no nível seguinte $i + 1$. Por exemplo, para um problema P com $D = \{a, b\}$, a floresta $G_D P$ seria

  $$\begin{array}{cccc} \varnothing & \{a\} & \{b\} & \{a,b\} \\ | & | & | & | \\ \varnothing & \{a\} & \{b\} & \{a,b\} \\ | & | & | & | \\ \varnothing & \{a\} & \{b\} & \{a,b\} \\ | & | & | & | \\ \vdots & \vdots & \vdots & \vdots \end{array}$$

Quanto aos morfismos, G_D mapeia uma redução $P \xrightarrow{(\tau,\sigma)} P'$ no homomorfismo $G_DP' \to G_DP$ que leva cada nó A' de G_DP' no nó $\tau^{-1}(A')$ do mesmo nível de G_DP, com A' um subconjunto de D', e

$$\tau^{-1}(A') = \{\ d \in D \mid \tau(d) \in A'\ \}$$

– Para representar os domínios de respostas, o funtor

$$G_R : \mathbf{Prob}_0 \to \mathbf{Floresta}$$

associa cada problema $P = \langle\ D, R, p\ \rangle$ a uma floresta cujo conjunto de nós em cada nível i é exatamente o conjunto de respostas de P, e cada nó tem como único filho a cópia de si mesmo no nível seguinte $i + 1$. Por exemplo, para um problema P com $R = \{c, d, e\}$, a floresta G_RP seria

$$\begin{array}{ccc} c & d & e \\ | & | & | \\ c & d & e \\ | & | & | \\ c & d & e \\ | & | & | \\ \vdots & \vdots & \vdots \end{array}$$

Quanto aos morfismos, G_R mapeia uma redução $P \xrightarrow{(\tau,\sigma)} P'$ no homomorfismo $G_RP' \to G_RP$ que leva cada nó r' de G_RP' no nó $\sigma(r')$ do mesmo nível de G_RP.

– Para representar as condições dos problemas, o funtor

$$G_p : \mathbf{Prob}_0 \to \mathbf{Floresta}$$

associa cada problema $P = \langle\ D, R, p\ \rangle$ a uma floresta cujo conjunto de nós em cada nível é exatamente o conjunto

$$\{\ (A, r) \mid A \neq \varnothing,\ \forall d \in A : (d, r) \in p\ \}$$

com $A \subseteq D$; ou seja, o conjunto acima está contido em $\wp(D) \times R$. Cada nó do nível i tem como único filho a cópia de si mesmo no nível seguinte $i + 1$. Por exemplo, para um problema

$$P = \langle\ \{a, b\},\ \{c, d, e\},\ \{(a, c), (a, d), (b, d), (b, e)\}\ \rangle$$

a floresta G_pP seria

$$\begin{array}{ccccc}
(\{a\},c) & (\{a\},d) & (\{b\},d) & (\{b\},e) & (\{a,b\},d) \\
| & | & | & | & | \\
(\{a\},c) & (\{a\},d) & (\{b\},d) & (\{b\},e) & (\{a,b\},d) \\
| & | & | & | & | \\
(\{a\},c) & (\{a\},d) & (\{b\},d) & (\{b\},e) & (\{a,b\},d) \\
| & | & | & | & | \\
\vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

Quanto aos morfismos, G_p mapeia uma redução $P \xrightarrow{(\tau,\sigma)} P'$ no homomorfismo $G_pP' \to G_pP$ que leva cada nó (A', r') de G_pP' no nó $(\tau^{-1}(A'), \sigma(r'))$ do mesmo nível de G_pP.

5.5 Definindo Estratégias de Busca

5.5.1 Busca em ECF. Baseados na discussão de 5.1, apresentamos uma definição de estratégia de busca no nosso *topos*. Em seguida, expressamos tal definição na lógica do *topos*.

5.5.2 Definição: estratégia de busca. Seja C o objeto

$$\coprod_{\langle\, T,\theta \,\rangle \,\in trees(\langle\, G,\lambda \,\rangle)} strings(\langle\, T,\theta \,\rangle)$$

de **ECF**, cujos elementos são todos os pares da forma

$$(\langle\, T,\theta \,\rangle, \langle\, H,\kappa \,\rangle)$$

com $\langle\, T,\theta \,\rangle$ uma árvore de $\langle\, G,\lambda \,\rangle$, e $\langle\, H,\kappa \,\rangle$ uma corda de $\langle\, T,\theta \,\rangle$.

Então, dado um objeto $\langle\, G,\lambda \,\rangle$ de **ECFR**, com G não- colapsante, uma estratégia de busca para $\langle\, G,\lambda \,\rangle$ é um par de morfismos em **ECF**

$$\langle\, init, step \,\rangle$$

onde

1. O morfismo
$$init : 1 \rightarrow C$$
escolhe um elemento de C da forma $(\langle\ H, \kappa\ \rangle, \langle\ H, \kappa\ \rangle)$, onde as duas componentes iguais representam uma corda raiz de $\langle\ G, \lambda\ \rangle$ – a corda inicial da busca;

2. O morfismo
$$step : C \rightarrow C$$
leva um par
$$(\langle\ T, \theta\ \rangle, \langle\ H, \kappa\ \rangle)$$
com $\langle\ H, \kappa\ \rangle$ uma corda de $\langle\ T, \theta\ \rangle$, a um par
$$(\langle\ T', \theta'\ \rangle, \langle\ H', \kappa'\ \rangle)$$
com $\langle\ H', \kappa'\ \rangle$ uma corda de $\langle\ T', \theta'\ \rangle$, tal que
   (a) $\langle\ T, \theta\ \rangle \subseteq \langle\ T', \theta'\ \rangle$;
   (b) $\langle\ T', \theta'\ \rangle$ tem, no máximo, uma corda a mais que $\langle\ T, \theta\ \rangle$.

A definição acima é análoga à representação do comportamento de uma busca sequencial em uma floresta por uma sequência de pares (ver 5.1.5) cuja primeira componente é uma árvore e cuja segunda componente é um nó da árvore.

5.5.3 Definindo estratégias de busca na lógica interna. Usando os predicados e termos definidos na seção anterior, podemos definir um predicado, na linguagem de **ECF**, que diz quando um termo da forma
$$\langle\ init, step\ \rangle$$
representa uma estratégia de busca em um objeto $\langle\ G, \lambda\ \rangle$ de **ECFR**. Aqui, *init* é um termo do tipo $\mathbf{C^1}$, e *step* é um termo do tipo $\mathbf{C^C}$, onde $\mathbf{C}$ é o objeto representado por
$$\coprod_{\langle\ T, \theta\ \rangle\ \in trees(\langle\ G, \lambda\ \rangle)} strings(\langle\ T, \theta\ \rangle)$$

$$
\boxed{
\begin{array}{l}
\mathbf{isSearchStrategy}(init, step, G, \lambda) \Leftrightarrow \\
\quad \mathbf{isNonCollapsing}(G) \wedge \\
\quad \pi_1(init(\star)) = \pi_2(init(\star)) \wedge \\
\quad \mathbf{isRoot}(\pi_2(init(\star)), G, \lambda) \wedge \\
\quad \forall(\langle\, T, \theta \,\rangle, \langle\, H, \kappa \,\rangle) \in C : \\
\quad \forall(\langle\, T', \theta' \,\rangle, \langle\, H', \kappa' \,\rangle) \in C : (\\
\qquad (((\langle\, T, \theta \,\rangle, \langle\, H, \kappa \,\rangle), (\langle\, T', \theta' \,\rangle, \langle\, H', \kappa' \,\rangle)) \in step \\
\qquad \Rightarrow \\
\qquad (\langle\, T, \theta \,\rangle \; = \; \langle\, T', \theta' \,\rangle \vee \\
\qquad\quad (\mathbf{extendsByOne}(T', \theta', T, \theta) \wedge \\
\qquad\quad \mathbf{wasAdded}(H', \kappa', T', \theta', T, \theta)) \\
\qquad) \\
\quad)
\end{array}
}
$$

Aqui, a segunda fórmula da conjunção diz que *init* de fato escolhe um par onde as duas componentes são iguais; a terceira fórmula afirma que cada componente representa uma corda raiz; a quarta diz que *step* define uma sequência satisfazendo as condições descritas na definição 5.5.2 acima.

5.5.4 O morfismo $stage_{\langle\, init, step \,\rangle}$. Considerando o exposto sobre o objeto de números naturais em 4.5, pode-se ver que uma estratégia de busca $\langle\, init, step \,\rangle$ para um objeto $\langle\, G, \lambda \,\rangle$ define, por recursão simples, um morfismo de N (o objeto de números naturais) para o objeto C definido em 5.5.2. Chamaremos este morfismo de $stage_{\langle\, init, step \,\rangle}$. Então o seguinte diagrama comuta:

$$
\begin{tikzcd}
& N \arrow[r, "s"] \arrow[dd, "stage_{\langle\, init, step \,\rangle}"] & N \arrow[dd, "stage_{\langle\, init, step \,\rangle}"] \\
1 \arrow[ur, "0"] \arrow[dr, "init"'] & & \\
& C \arrow[r, "step"'] & C
\end{tikzcd}
$$

O morfismo $stage_{\langle\, init, step \,\rangle}$ pode ser visto como análogo a uma função que fornece, para cada natural n, a sub-árvore de $\langle\, G, \lambda \,\rangle$ composta pelas cordas visitadas até o passo n da busca e a corda visitada no passo n da busca.

Quando não houver risco de confusão, chamaremos este morfismo simplesmente de *stage*. Usaremos o mesmo nome para nos referirmos ao termo *stage*, do tipo $\mathbf{C}^{\mathbf{N}}$, que representa este morfismo na linguagem interna de **ECF**.

5.6
Definindo Heurísticas

5.6.1 Definição: heurística. Agora estamos em condições de definir heurísticas na lógica do nosso *topos*. Uma heurística é representada por uma quádrupla

$$\langle\, G, \lambda, init, step \,\rangle$$

satisfazendo $\mathbf{isSearchStrategy}(init, step, G, \lambda)$.

5.6.2 Respostas retornadas por uma heurística. Uma estratégia de busca cuja execução termine deve retornar uma resposta. De acordo com nossas definições, isto ocorre quando é encontrado um ponto fixo do morfismo *step*; i.e., quando iterações subsequentes da busca não alteram mais a árvore de cordas visitadas até então, passando a revisitar a mesma corda v indefinidamente. Esta corda v corresponderá à resposta retornada pela busca. Isto nos motiva a definir o predicado

$$\boxed{\begin{array}{l} \mathbf{isReturned}(H, \kappa, G, \lambda, init, step) \Leftrightarrow \\ \quad \langle\, H, \kappa \,\rangle \in strings(G, \lambda) \wedge \\ \quad \exists n \in N : \\ \quad \exists \langle\, T, \theta \,\rangle \in trees(G, \lambda) : (\\ \quad\quad stage(n) = (\langle\, T, \theta \,\rangle, \langle\, H, \kappa \,\rangle) \wedge \\ \quad\quad stage(sn) = stage(n) \\ \quad) \end{array}}$$

com κ do tipo $\mathbf{L^H}$, λ do tipo $\mathbf{L^G}$, *init* do tipo $\mathbf{C^1}$, e *step* do tipo $\mathbf{C^C}$, para dizer que $\langle\, H, \kappa \,\rangle$ é a corda retornada pela estratégia de busca $\langle\, init, step \,\rangle$ quando executada sobre o objeto $\langle\, G, \lambda \,\rangle$. No predicado, N é o objeto de números naturais, $s : N \to N$ é o morfismo sucessor, e *stage* é o morfismo induzido por $\langle\, init, step \,\rangle$, descrito em 5.5.4.

5.6.3 Uma estratégia exata de construção de florestas de respostas. Para podermos verificar se uma heurística retorna respostas corretas para os problemas de $\mathbf{Prob}_0$, definiremos uma estratégia “exata” de construção de florestas de respostas $\langle\ E, \varepsilon\ \rangle$, descrita a seguir:

Para cada problema P, a floresta EP é uma floresta composta por uma quantidade infinita de árvores lineares finitas. Lembre-se de que todo problema $\langle\ D, R, p\ \rangle$ em $\mathbf{Prob}_0$ tem uma instância d escolhida pela única redução $P_1 \longrightarrow P$ em $\mathbf{Prob}_0$ (ver 3.3.5). Então, a floresta EP consiste de todas as árvores lineares finitas rotuladas da forma

$$(A_0, n_0) \text{——} (A_1, n_1) \text{——} \quad \cdots \quad \text{——} (\{r\}, n_k)$$

com cada A_i um conjunto de respostas de R, cada n_i um natural, e r uma resposta correta para d (i.e., $(d, r) \in p$). A idéia é que exista uma árvore em EP para cada caminho possível (de qualquer comprimento finito) terminando em um nó folha rotulado por uma resposta correta e algum natural.

Quanto aos morfismos, a estratégia $\langle\ E, \varepsilon\ \rangle$ mapeia uma redução $P \xrightarrow{(\tau,\sigma)} P'$ de $\mathbf{Prob}_0$ no morfismo $EP' \longrightarrow EP$ que leva cada nó (A'_i, n'_i) de EP' no nó $(\sigma(A'_i), n'_i)$ de EP.

5.6.4 Verificando se uma heurística retorna respostas exatas. Uma vez definida a estratégia exata $\langle\ E, \varepsilon\ \rangle$, podemos usá-la para verificar se uma dada heurística $\langle\ G, \lambda, \mathit{init}, \mathit{step}\ \rangle$ retorna respostas corretas. Basta verificar se a corda $\langle\ H, \kappa\ \rangle$ retornada pela heurística é (isomorfa a) alguma corda folha de $\langle\ E, \varepsilon\ \rangle$. Se isto acontecer, então, para todo problema P de $\mathbf{Prob}_0$, o nó definido por HP é rotulado por uma resposta correta para a instância escolhida d de P. Isto é descrito pelo predicado

$$\boxed{\begin{array}{l} \mathbf{isExact}(G, \lambda, \mathit{init}, \mathit{step}) \Leftrightarrow \\ \quad \exists \langle\ H, \kappa\ \rangle\ \in \mathit{strings}(G, \lambda) : (\\ \quad\quad \mathbf{isReturned}(H, \kappa, G, \lambda, \mathit{init}, \mathit{step}) \wedge \\ \quad\quad \mathbf{isLeaf}(H, \kappa, E, \varepsilon) \\ \quad) \end{array}}$$

com λ do tipo $\mathbf{L^G}$, *init* do tipo $\mathbf{C^1}$, e *step* do tipo $\mathbf{C^C}$

5.6.5 Extraindo valores de uma corda. Para especificar uma heurística, pode ser necessário lidar com os valores naturais associados a cada corda (i.e., os valores que representam a qualidade de cada nó).

Lembre-se de que toda corda $\langle\ H, \kappa\ \rangle$ de um objeto $\langle\ G, \lambda\ \rangle$ define um nó v da floresta GP_1, onde $P_1 = \langle\ \{\star\}, \{\star\}, \{(\star, \star\}\ \rangle$ é objeto inicial de $\mathbf{Prob}_0$ (fato que faz com que haja um homomorfismo de toda floresta GP para a floresta GP_i). Tomaremos o valor da corda $\langle\ H, \kappa\ \rangle$ como sendo a nota do nó v.

Podemos dividir o processo de obter a nota do nó v em duas partes: primeiro, obter o nó $\kappa(v)$ de LP_i; em seguida, obter o valor natural associado a $\kappa(v)$. Com isto em mente, definiremos o morfismo

$$N \xrightarrow{\nu} PPL$$

onde N é o objeto de números naturais de **ECF** e PPL é o objeto de sub-objetos de sub-objetos de L. O morfismo ν deve associar cada natural $n \in N$ ao sub-objeto composto por cordas de L que têm valor n.

Com este morfismo, podemos definir o termo

$$\boxed{grade(H, \kappa, G, \lambda) = n}$$

do tipo $\mathbf{N}$, onde $\langle\ H, \kappa\ \rangle$ é uma corda de $\langle\ G, \lambda\ \rangle$, e n é o natural tal que

$$P\kappa(H) \in \nu(n)$$

onde $P\kappa$ é o morfismo

$$\begin{array}{rcl} P\kappa : PH & \rightarrow & PL \\ J & \mapsto & \{\kappa(j) \mid j \in J\} \end{array}$$

5.7
Conclusões do Capítulo

5.7.1 Resumo. Este capítulo apresentou uma definição de estratégia de busca no nosso modelo e traduziu tal definição para a linguagem de teoria local dos conjuntos – mais especificamente, para a teoria do *topos* **ECF**. No próximo capítulo, veremos exemplos de heurísticas especificadas nesta teoria, além de outras aplicações do nosso modelo.