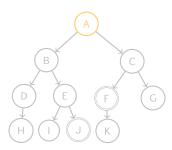
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do | Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])]; n1 <- successeur(n); Ch Premier(Liste);

ſ	Ch	n	n1	fin	Liste	Sol
ĺ	-	-	-	-	{{A}}	



Ch <- Premier(Liste); fin <- Faux;

while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch); n1 <- successeur(n):

while non(fin) et n1 est valide do

if n1 est solution then

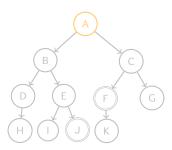
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	-	-	Faux	{{ A }}	



Ch <- Premier(Liste); fin <- Faux;

while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);

n1 <- successeur(n);

while non(fin) et n1 est valide do

if n1 est solution then

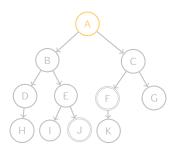
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

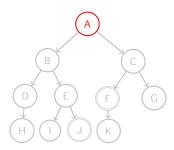
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	-	-	Faux	{{ A }}	



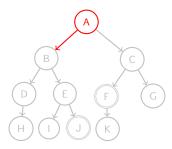
Ch <- Premier(Liste); fin <- Faux;
while Ch!= || et non(fin) do
Liste <- Liste / |Ch|; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf |n1|; fin <- Vrai;
else Liste <- Liste Uf |(Ch Uf |n1|)|;
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	-	Faux	{}	



$$\begin{split} & \text{Ch} <\text{-Premier(Liste)}; & \text{fin} <\text{-Faux}; \\ & \text{while Ch!} = || \text{ et non(fin) do} \\ & \text{Liste} <\text{-Liste} / |\text{Ch}|; & \text{n} <\text{-dernierNoeud(Ch)}; \\ & \text{n1} <\text{-successeur(n)}; \\ & \text{while non(fin) et n1 est valide do} \\ & \text{if n1 est solution then} \\ & \text{Sol Ch UF [n1]; fin} <\text{-Vrai}; \\ & \text{else Liste} <\text{-Liste UF [(Ch Uf [n1])]}; \\ & \text{n1} <\text{-successeur(n)}; \\ & \text{Ch Premier(Liste)}; \end{split}$$

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);

n1 <- successeur(n);

while non(fin) et n1 est valide do

if n1 est solution then

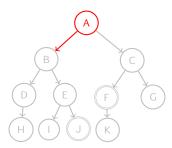
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

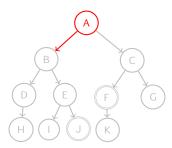
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



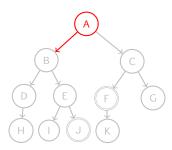
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / |Ch|; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf |n1|; fin <- Vrai; else Liste <- Liste Uf |(Ch Uf |n1|)|; n1 <- successeur(n); Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{}	



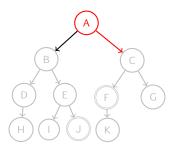
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	В	Faux	{{AB}}	



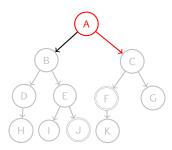
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n); Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	С	Faux	{{AB}}	



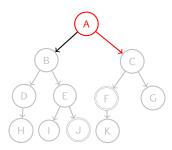
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

n	n1	fin	Liste	Sol
Α	С	Faux	{{AB}}	
	n A	A C		



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste);

n	n1	fin	Liste	Sol
Α	С	Faux	{{AB}}	
	n A	Λ (



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n);

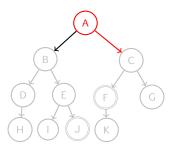
while non(fin) et n1 est valide do

if n1 est solution then Sol Ch Uf {n1}; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

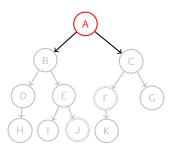
n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	С	Faux	{{AB},{AC}}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	Т	Faux	{{AB},{AC}}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n);

while non(fin) et n1 est valide do

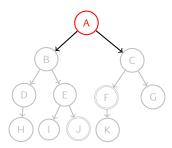
if n1 est solution then

 $Sol\ Ch\ Uf\ \{n1\}; fin<-\ Vrai;$

else Liste <- Liste Uf $\{(Ch\ Uf\ \{n1\})\};$

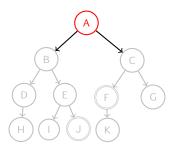
 $n1 \leftarrow successeur(n);$

1	Ch	n	n1	fin	Liste	Sol
	{A}	Α	1	Faux	{{AB},{AC}}	
Į						



$$\label{eq:Characteristics} \begin{split} &\text{Ch} <\text{- Premier(Liste)}; &\text{fin} <\text{- Faux}; \\ &\text{while Ch!= }[] \text{ et non(fin) do} \\ &\text{Liste } <\text{- Liste } / [\text{Ch}]; n <\text{- dernierNoeud(Ch)}; \\ &n1 <\text{- successeur(n)}; \\ &\text{while non(fin) et n1 est valide do} \\ &\text{if n1 est solution then} \\ &\text{Sol Ch Uf }[n1]; &\text{fin} <\text{- Vrai}; \\ &\text{else Liste } <\text{- Liste Uf }[(\text{Ch Uf }[n1])]; \\ &n1 <\text{- successeur(n)}; \end{split}$$

Ch	n	n1	fin	Liste	Sol
{A} {AB}	Α		Faux	{{AB},{AC}}	



Ch <- Premier(Liste); fin <- Faux;

while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);

n1 <- successeur(n);

while non(fin) et n1 est valide do

if n1 est solution then

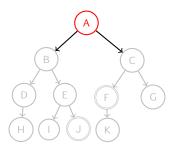
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

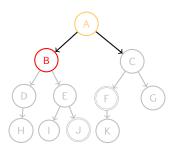
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A} {AB}	Α		Faux	{{AB},{AC}}	



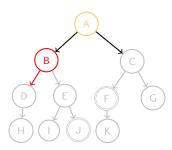
Ch <- Premier(Liste); fin <- Faux;
while Ch!= || et non(fin) do
Liste <- Liste / |Ch|; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf |n1|; fin <- Vrai;
else Liste <- Liste Uf |(Ch Uf |n1|)|;
n1 <- successeur(n);
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	1	Faux	{{AB},{AC}} { <mark>AC</mark> }	



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

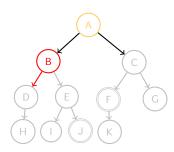
Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	D D	Faux	{{AB},{AC}} {AC}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / {Ch}; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai;

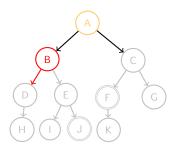
else Liste <- Liste Uf {(Ch Uf {n1}));
n1 <- successeur(n);
Ch Premier(Liste);
return fin;</pre>

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	D		{AC}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste);

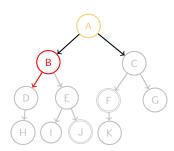
Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	D	Faux	{{AB},{AC}} {AC}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do | Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1]));

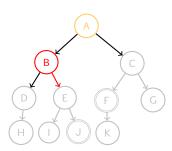
n1 <- successeur(n);
Ch Premier(Liste);
return fin;</pre>

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	D D	Faux	{{AB},{AC}} {{AC},{ABD}}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

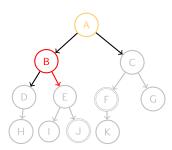
Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	E	Faux	{{AB},{AC}} {{AC},{ABD}}	



Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do | Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])];

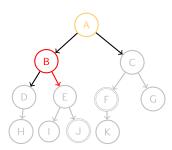
n1 <- successeur(n);
Ch Premier(Liste);
return fin;</pre>

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	E	Faux	{{AB},{AC}} {{AC},{ABD}}	



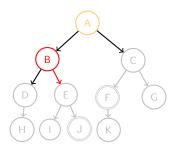
Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	E E	Faux	{{AB},{AC}} {{AC},{ABD}}	



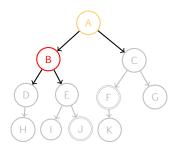
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A} {AB}	A B	E	Faux	{{AB},{AC}} {{AC},{ABD},{ABE}}	



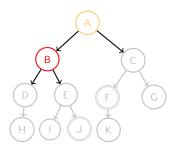
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

ſ	Ch	n	n1	fin	Liste	Sol
	{A} {AB}	A B		Faux	{{AB},{AC}} {{AC},{ABD},{ABE}}	



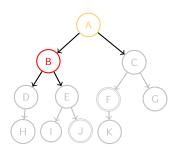
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Γ	Ch	n	n1	fin	Liste	Sol
	{A} {AB}	A B	T T	Faux	{{AB},{AC}} {{AC},{ABD},{ABE}}	



$$\label{eq:Characteristics} \begin{split} &\text{Ch} <\text{- Premier(Liste)}; &\text{fin} <\text{- Faux}; \\ &\text{while Ch!= }[] \text{ et non(fin) do} \\ &\text{Liste } <\text{- Liste } / [\text{Ch}]; n <\text{- dernierNoeud(Ch)}; \\ &n1 <\text{- successeur(n)}; \\ &\text{while non(fin) et n1 est valide do} \\ &\text{if n1 est solution then} \\ &\text{Sol Ch Uf }[n1]; &\text{fin} <\text{- Vrai}; \\ &\text{else Liste } <\text{- Liste Uf }[(\text{Ch Uf }[n1])]; \\ &n1 <\text{- successeur(n)}; \end{split}$$

Ch	n	n1	fin	Liste	Sol
{A} {AB} {AC}	A B	⊥ ⊥	Faux	{{AB},{AC}} {{AC},{ABD},{ABE}}	



Ch <- Premier(Liste); fin <- Faux;

while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);

n1 <- successeur(n);

while non(fin) et n1 est valide do

if n1 est solution then

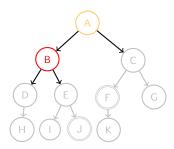
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

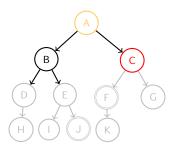
Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В			{{AC},{ABD},{ABE}}	
{AC}					



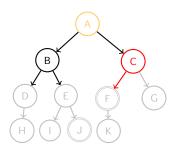
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / {Ch}; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf {(Ch Uf [n1])}; n1 <- successeur(n); Ch Premier(Liste);

Ch	n	n1	fin	Liste	Sol
{A}	Α	Т	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С			{{ABD},{ABE}}	



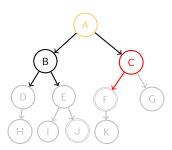
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



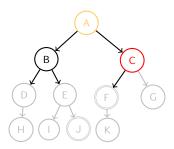
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



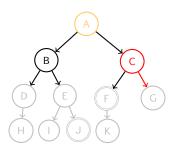
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Ch	n	n1	fin	Liste	Sol
{A}	Α	1	Faux	{{AB},{AC}}	
{AB}	В	1		{{AC},{ABD},{ABE}}	
{AC}	С	F		{{ABD},{ABE}}	



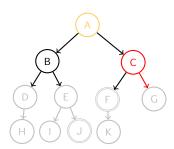
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do | Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch UF [n1]; fin <- Vrai; else Liste <- Liste UF [(Ch UF [n1])]; n1 <- successeur(n); Ch Premier(Liste);

Sol
,C,F



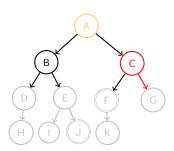
Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Sol
۹,C,F }



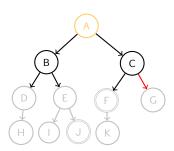
Ch <- Premier(Liste); fin <- Faux; while Ch!= {| et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

Sol
{A,C,F}
_



Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

	Ch	n	n1	fin	Liste	Sol
ĺ	{A}	Α	1	Faux	{{AB},{AC}}	
	{AB}	В	1		{{AC},{ABD},{ABE}}	
	{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



Ch <- Premier(Liste); fin <- Faux;

while Ch!= {} et non(fin) do

Liste <- Liste / {Ch}; n <- dernierNoeud(Ch);

n1 <- successeur(n):

while non(fin) et n1 est valide do

if n1 est solution then

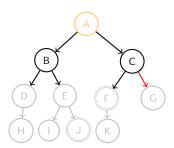
Sol Ch Uf $\{n1\}$; fin <- Vrai;

else Liste \leftarrow Liste Uf $\{(Ch Uf \{n1\})\};$

n1 <- successeur(n);

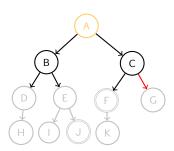
Ch Premier(Liste);

	Ch	n	n1	fin	Liste	Sol
ĺ	{A}	Α	1	Faux	{{AB},{AC}}	
ı	{AB}	В	1		{{AC},{ABD},{ABE}}	
	{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}
•		•	•			



Ch <- Premier(Liste); fin <- Faux;
while Ch!= [] et non(fin) do
Liste <- Liste / [Ch]; n <- dernierNoeud(Ch);
n1 <- successeur(n);
while non(fin) et n1 est valide do
if n1 est solution then
Sol Ch Uf [n1]; fin <- Vrai;
else Liste <- Liste Uf [(Ch Uf [n1])];
n1 <- successeur(n);

	Ch	n	n1	fin	Liste	Sol
	{A}	Α	1	Faux	{{AB},{AC}}	
ı	{AB}	В	1		{{AC},{ABD},{ABE}}	
	{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}



Ch <- Premier(Liste); fin <- Faux; while Ch!= || et non(fin) do Liste <- Liste / [Ch]; n <- dernierNoeud(Ch); n1 <- successeur(n); while non(fin) et n1 est valide do if n1 est solution then Sol Ch Uf [n1]; fin <- Vrai; else Liste <- Liste Uf [(Ch Uf [n1])); n1 <- successeur(n);

ſ	Ch	n	n1	fin	Liste	Sol
ſ	{A}	Α	1	Faux	{{AB},{AC}}	
l	{AB}	В	1		{{AC},{ABD},{ABE}}	
	{AC}	С	G	Vrai	{{ABD},{ABE}}	{A,C,F}
_						

