Construction of isomorphic classes of linear Binary codes

Based on Reseach By Harald Fripertinger, Adalbert Kerber and Reinhard Laue,

Presentation by:
Alexander Erick Trofimoff
PhD program
ECE Dept
Drexel U
2015 Spring

Bibliography

Betten A., Braun M., Fripertinger H., Kerber A., Kohnert A.,, Wasserman A., Error Correcting Linear Codes – Classification by Isometry and Applications , Springer Algorithmics and Computations in Mathematics Vol 18, 2006

Presentation Outline

- ✓ **Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)**
- Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.
- Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)
- Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.
- Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$
- General Algorithm of snakes and ladders for generation of codes.

Snakes and ladders algorithm
(Leiterspiel , B. Schmalz, 1992)

- ✓ Orderly generating of codes algorithm and its variation with canonical augmentation rely on the fact that we are able to compute a canonical from every subset, which is not always feasible.

- ✓ Computing the canonical form depends of the nature of the group action under consideration.

- ✓ **Snakes & Ladders** is an orbit algorithm which is general , it doesn't depend on the nature of the group.

- ✓ The cost of it is the amount of memory required correlates linearly to the number of orbits computed.

- ✓ The speedup from the memory vs time tradeoff makes it realistic to tackle instances of hard problems as computation of isometry classes of linear codes.

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

There two ways to describe the algorithm:

- Computing orbits of a group G on orbits on subsets of a set X on which G acts, or Double cosets in finite groups.

- The algorithm works along a sequence of subgroups which are alternatively subgroups and overgroups of each other (down and up process)

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

1. Assuming that the Orbits on points can be computed, the main goal is to provide a triple (τ, σ, φ) which is a solution to the orbit problem on G acting on admissible subsets of X.

2. **Inductive approach:** we are going to compute orbits of G on $\wp_i^{(f)}(X)$ for i=0,1,…., this corresponds to a **breath first search**.

3. If i=1 this is reduced to be the basic **Orbit on Points Algorithm**.

4. Let orbit $(G, \wp_i^{(f)}(X)) = (\tau_i, \sigma_i, \varphi_i)$ be a solution of the orbit problem on i-subsets.

5. Assume that a **transversal τ_i** of orbits of G on sets $\wp_i^{(f)}(X)$ has been already computed.

   A set R is canonical if it belongs to one of the transversals τ_i for some i.

6. In order to **Compute τ_{i+1}** , consider extensions of the sets in τ_i

   An extension is a set of the form $R \cup \{x\} \in \wp_{i+1}^{(f)}(X)$

   $R \subseteq \tau_i \subseteq \wp_i^{(f)}(X)$, $x \in X \backslash S$

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

7. To compute the next level of orbits on **(i+1)-sets** we have to deal with:

Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.

Problem 2: Determine when 2 extensions are isomorphic $\mathrm{R} \cup \{x\}$ and $\mathrm{S} \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)

Problem 3: Compute in y the Stabilizer $G_{\mathrm{R} \cup \{x\}}$, assuming that G_{R} is known.

Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$

Presentation outline

- Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)
- ✓ **Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.**
- Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)
- Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.
- Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$
- General Algorithm of snakes and ladders for generation of codes.

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

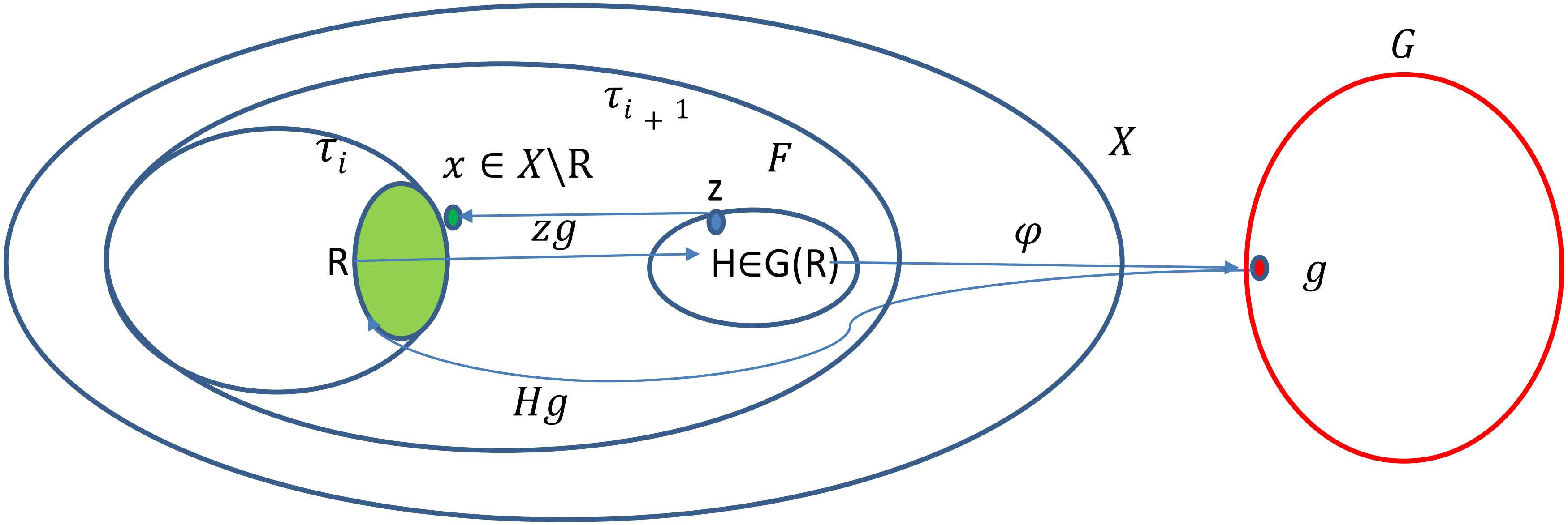
Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.

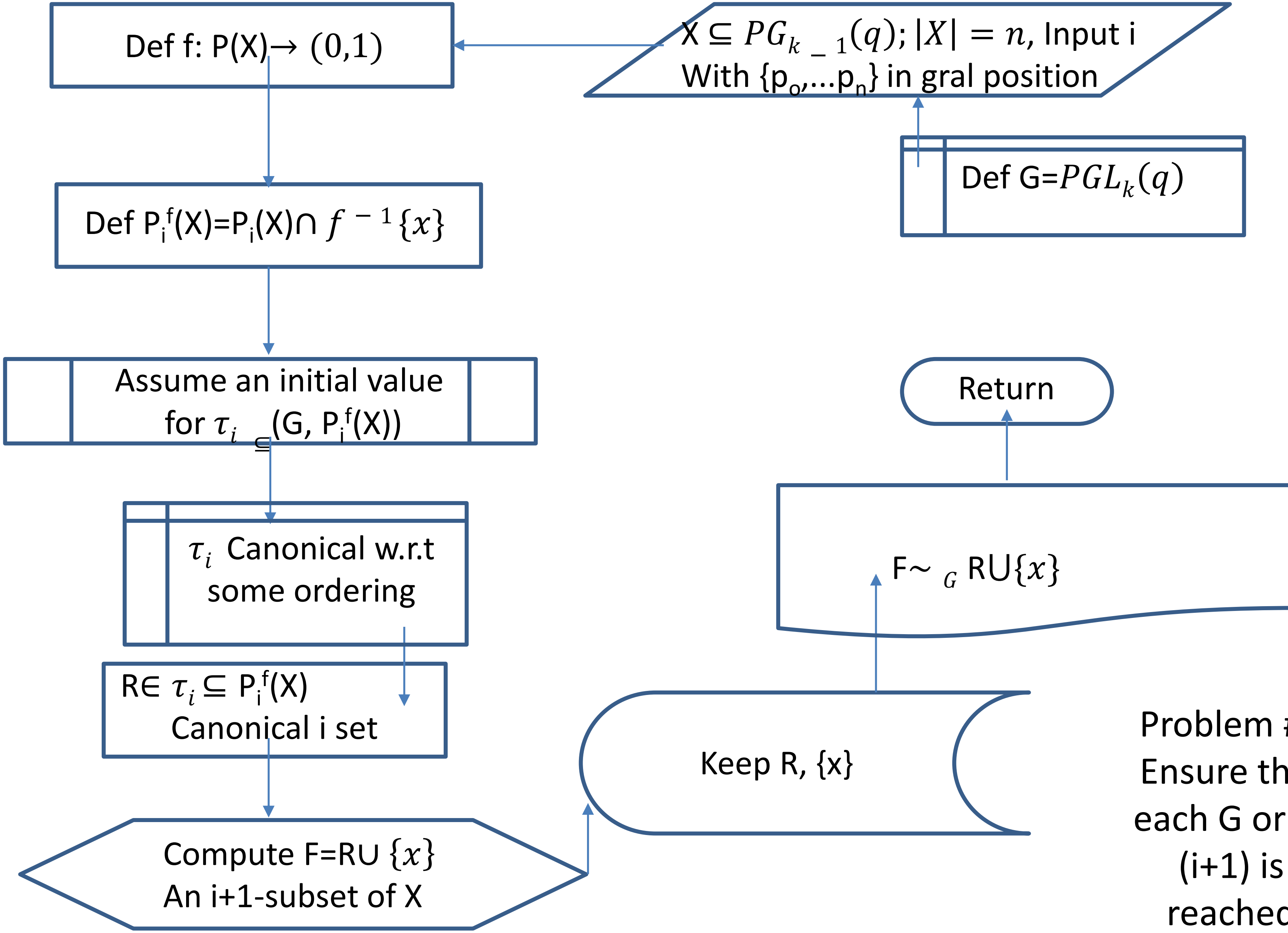
Let F be an admissible (i+1) –subset of X .

Let z:= max F , let H := $F\setminus\{z\}$, admissible since f is hereditary

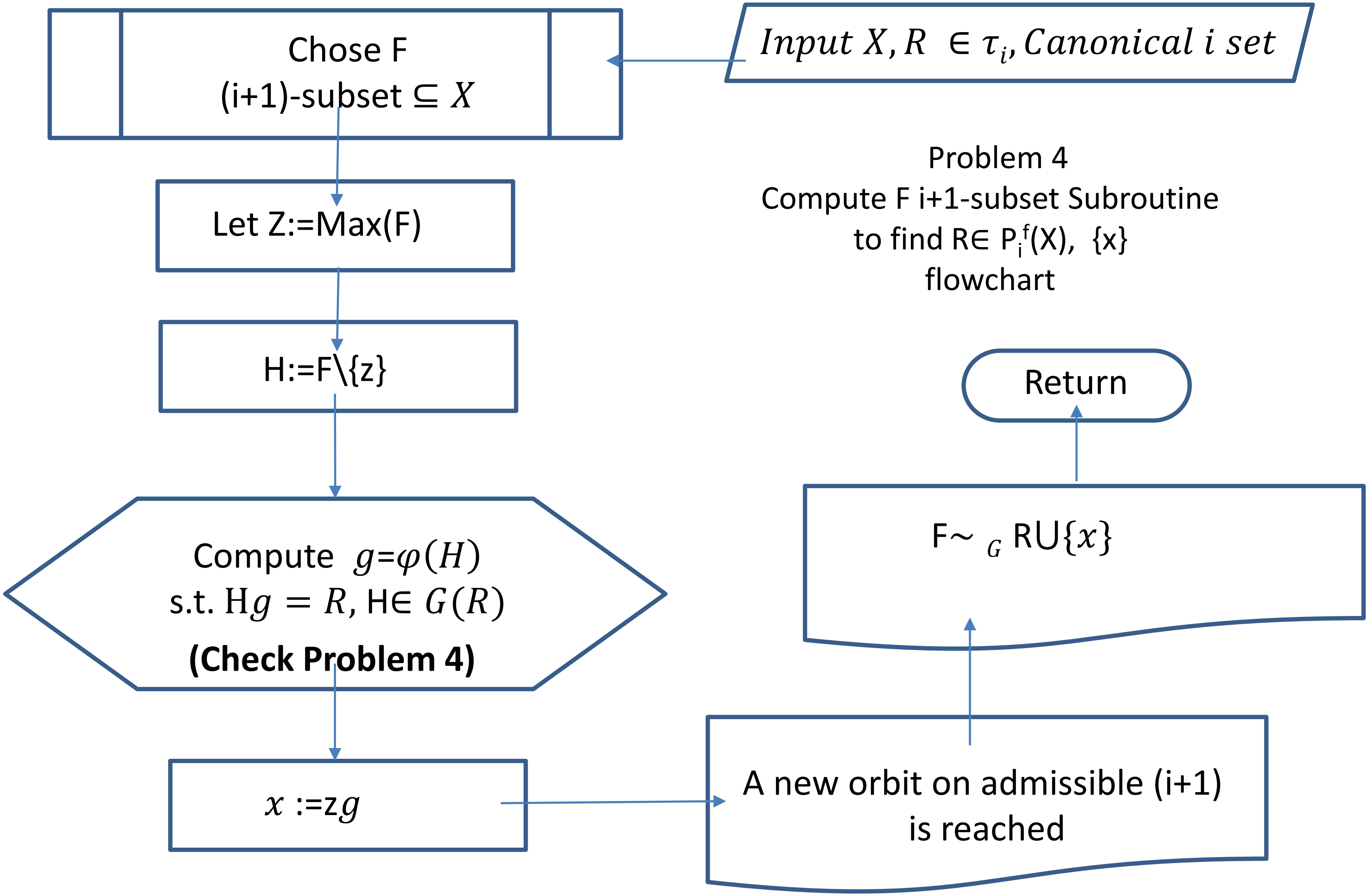
Thus $H \in G(R)$, for some $R \in \tau_i$, $Hg = R$, for some $g = \varphi(H)$

Let $x := zg \in X \setminus R$, therefore $F \sim_G R \cup \{x\}$, one of the candidate sets which We considered.





Problem #1
Ensure that
each G orbit
(i+1) is
reached
subroutine
flow chart



Problem 4
Compute F i+1-subset Subroutine
to find R∈ $P_i^f(X)$, {x}
flowchart

Presentation Outline

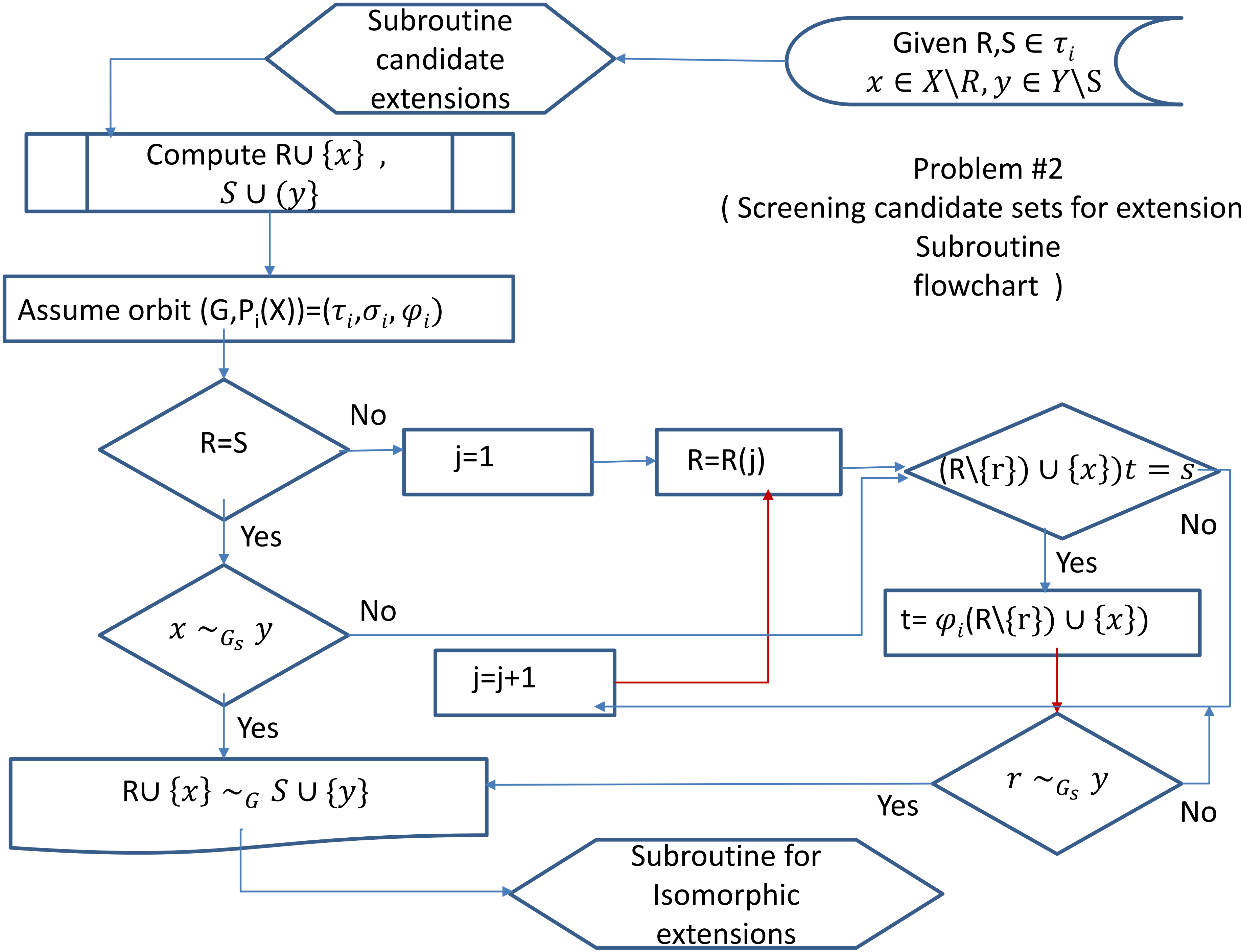
- Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)
- Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.
- ✓ **Problem 2: Determine when 2 extensions are isomorphic $\boldsymbol{R} \cup \{\boldsymbol{x}\}$ and $\boldsymbol{S} \cup \{\boldsymbol{y}\}$, i.e. belong to the same $\boldsymbol{G}$ -orbit. (R and S are both canonical, since $\boldsymbol{R}, \boldsymbol{S} \in \boldsymbol{\tau_i}$)**
- Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.
- Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$
- General Algorithm of snakes and ladders for generation of codes.

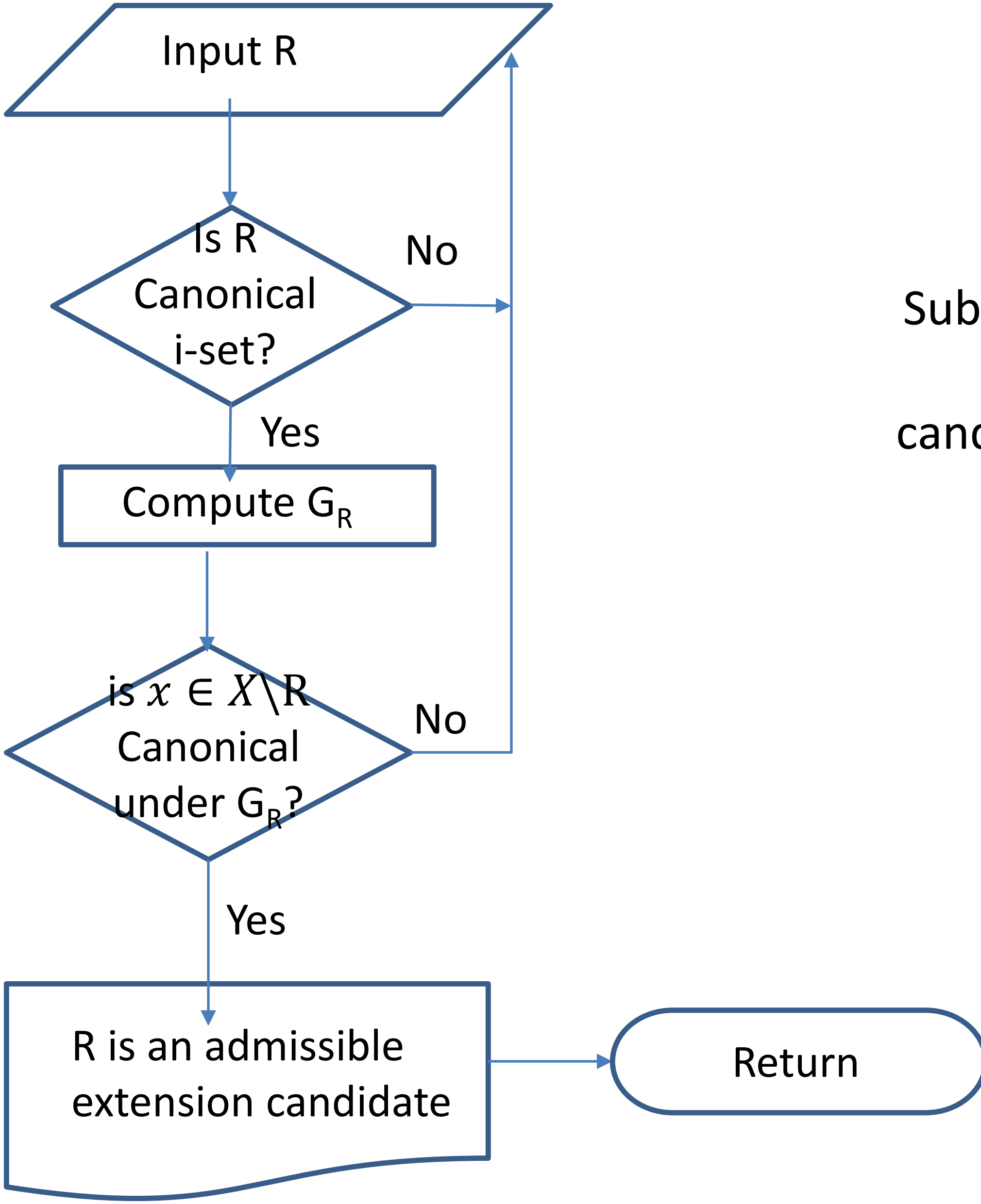
Why stabilizers subgroup are so important here?

A group action is *transitive* if $G \cdot S = S$. In other words, $\forall\, s,t \in S$, $\exists\, g \in G$ s.t. $g \cdot s = t$.
Equivalently, S contains a single orbit.

The *stabilizer* of s is the set $G_s = \{g \in G \mid g \cdot s = s\}$, the set of elements of G which leave s unchanged under the action.

The stabilizer G_s of any element $s \in S$ is a subgroup of G.


Subroutine candidate extensions
Given R,S ∈ τ_i
$x \in X\backslash R, y \in Y\backslash S$
Compute RU $\{x\}$, $S \cup \{y\}$
Problem #2
(Screening candidate sets for extension
Subroutine
flowchart)
Assume orbit (G,P$_i$(X))=(τ_i,σ_i, φ_i)
R=S
No
j=1
R=R(j)
(R\{r}) ∪ $\{x\})t = s$
No
Yes
Yes
$x \sim_{G_S} y$
No
t= φ_i(R\{r}) ∪ $\{x\}$)
j=j+1
Yes
$r \sim_{G_S} y$
RU $\{x\} \sim_G S \cup \{y\}$
Yes
No
Subroutine for Isomorphic extensions



Problem 2
Subroutine checking admissible candidate extensions flow chart

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G-orbit.

A necessary and sufficient condition is given by:

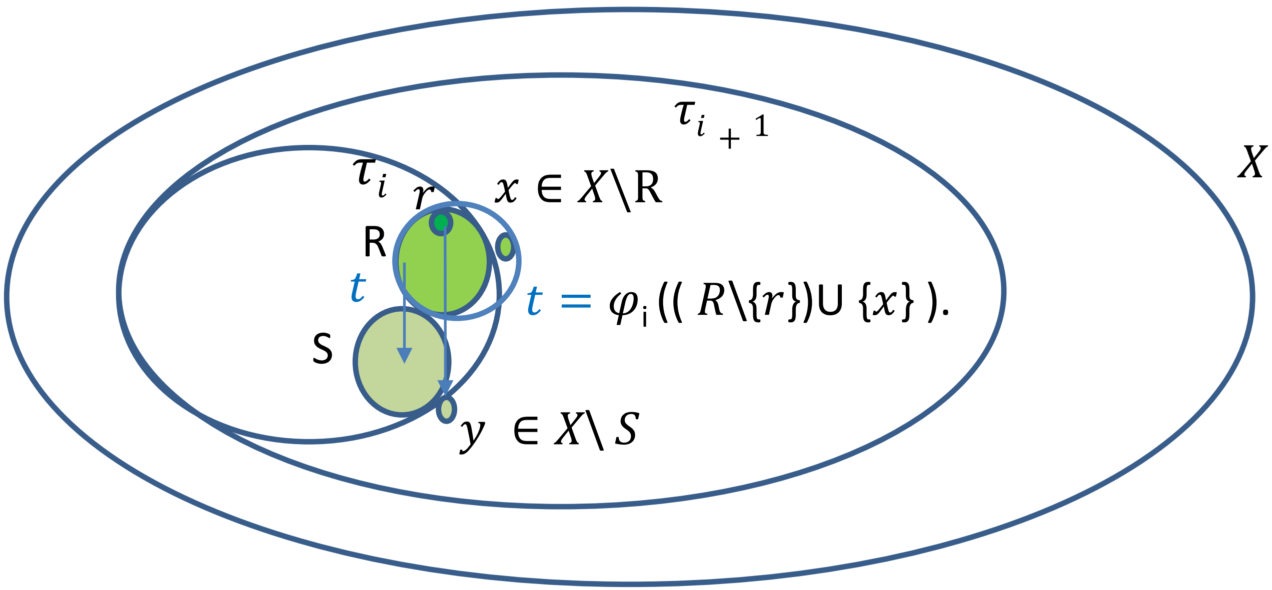
Lemma

Assume that the orbit $(G, \wp_i(X)) = (\tau_i, \sigma_i, \varphi_i)$, for $R, S \in \tau_i$, $x \in X \setminus R$, $y \in X \setminus S$,

We have $R \cup \{x\} \sim_G S \cup \{y\}$ iff one of the following conditions hold,

1. $R = S$ and $x \sim_{G_S} y$, or
2. $\exists\ r \in R$, s.t. $((R \setminus \{r\}) \cup \{x\})t = S$ and $rt \sim_{G_S} y$ where $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$.

Problem 2: Determine when 2 extensions are isomorphic



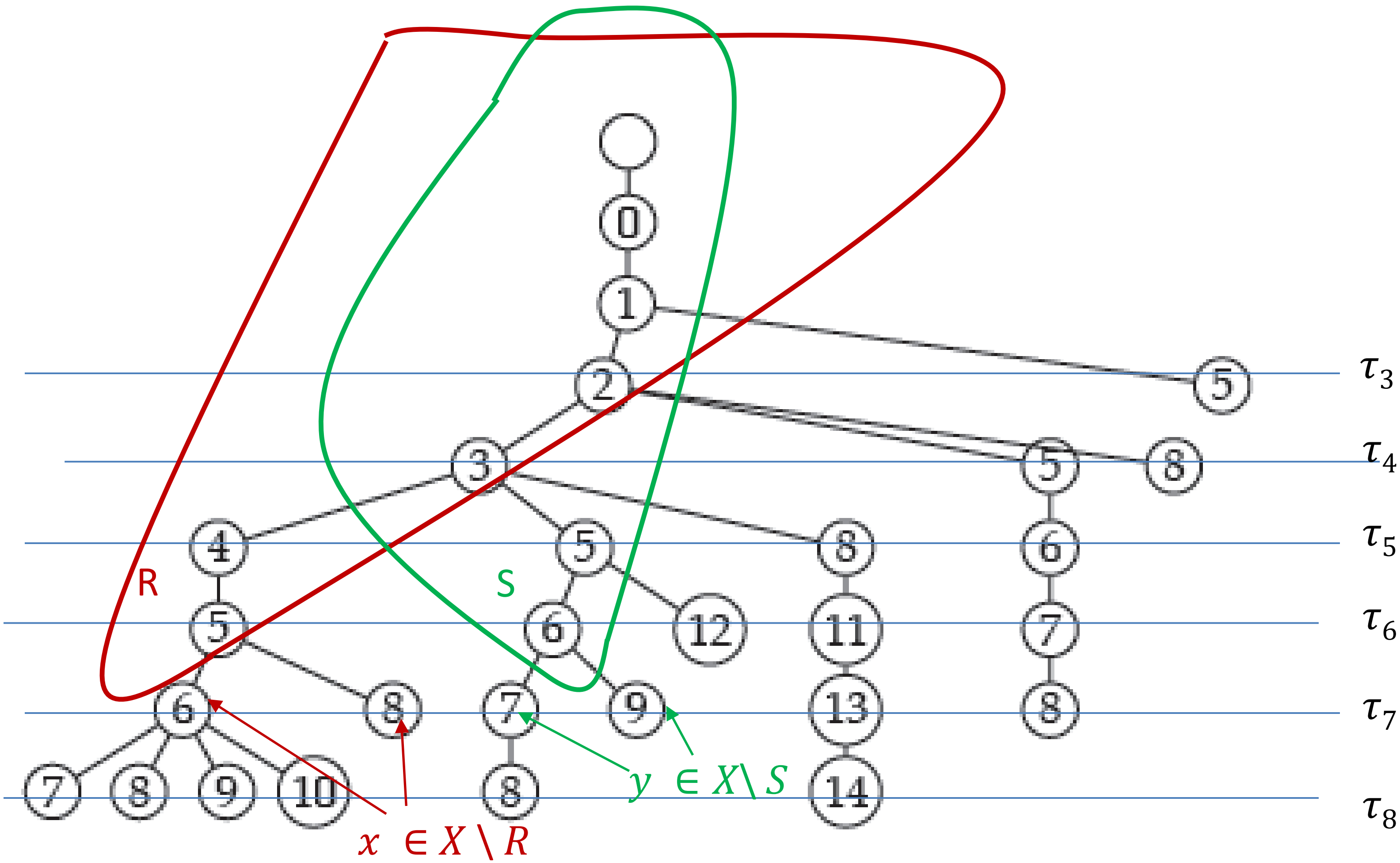


Fig. 9.4 Orbits of $\mathrm{PGL}_4(2)$ on $\mathcal{P}_{\leq 8}(\mathrm{PG}_3(2))$

Necessity

Case 1

$R \cup \{x\} \to_G S \cup \{y\}$

$R.h = S \quad x.h = y$

$R = S \quad R, S \in \tau_i$

$R.h = S.h = S$

$h \in G_s$

Case 2

$R \neq Rg \subseteq S \cup \{y\}$

$x.g \in S, \quad g \in (G,*)$

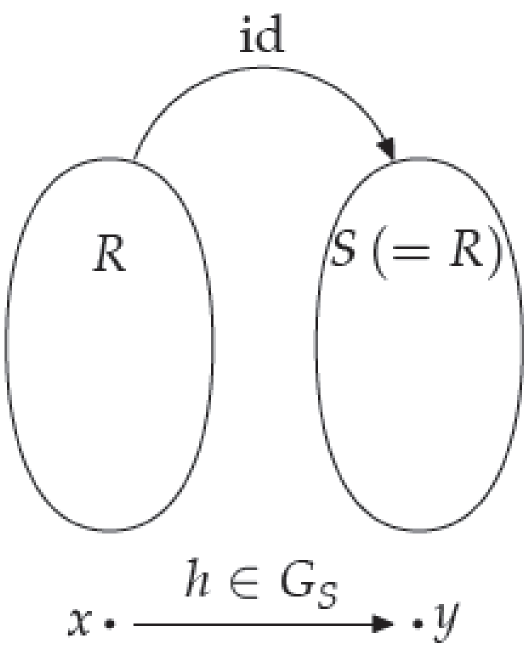
If $r = y.g^{-1}$

$g = t * h, \quad h \in (G_s,*)$

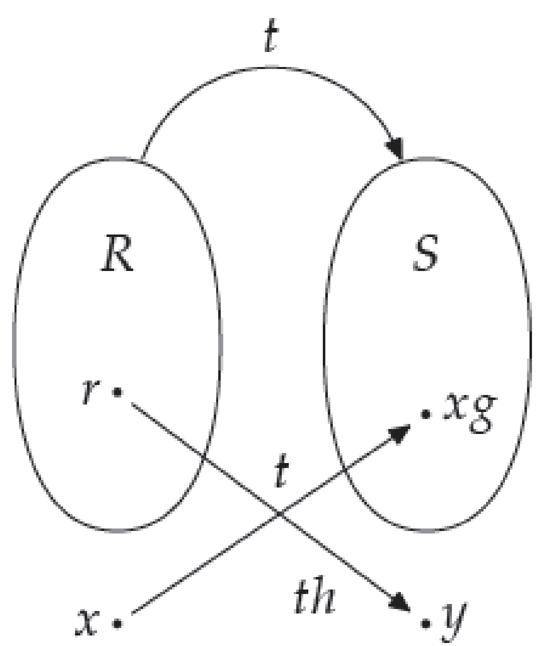
$g \in tG_s$

$(r).t.h = (r).(t * h) = r.g = y \quad ,$

$h \in G_s$



$x.h = y = x$



$(r)t \xrightarrow{G_s} y$

$t = \varphi_i((R\{r\}) \cup \{x\})$

Sufficiency

Case 1

$$R = S$$

$$x \to_{Gs} y$$

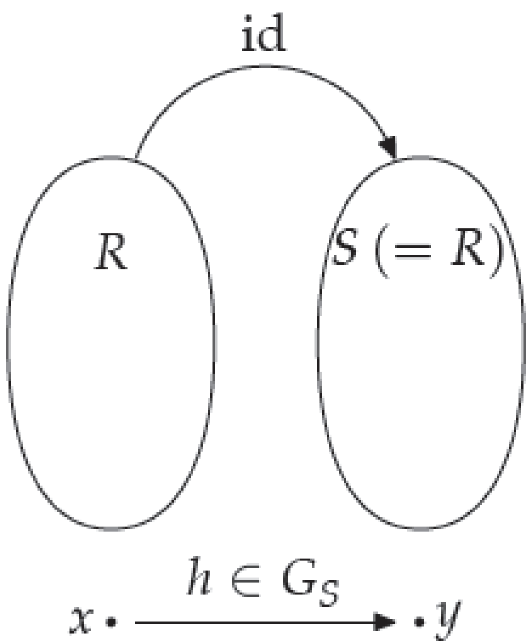
$$h \in Gs$$

$$x.h = y = x$$

$$(R \cup \{x\}).g$$

$$= R.g \cup \{x.g\}$$

$$= S \cup \{y\}$$



Case 2

$$\exists r \ \ s.t. \ \left((R\backslash\{r\}) \cup \{x\}\right).t = S$$

r.t.h=r.(t*h)=y, $h \in Gs$

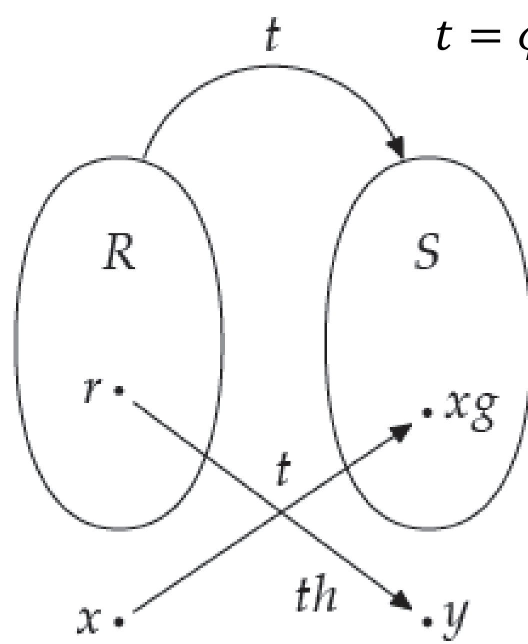
$$\left((R) \cup \{x\}\right).t.h$$

$$= \ \left((R\backslash\{r\}) \cup \{x\}\right).t.h \ \cup \{r\}.h$$

$$= S.h \ \cup \{r.t.h\}$$

$$= S \ \cup \{y\}$$

$$t = \varphi_i \left((R\{r\}) \cup \{x\}\right)$$



$$(r).t.h = r.g = \ y \quad , h \in (G_s, *)$$

$$(r)t \xrightarrow[G_s]{} y$$

Rejecting Isomorphic Extensions

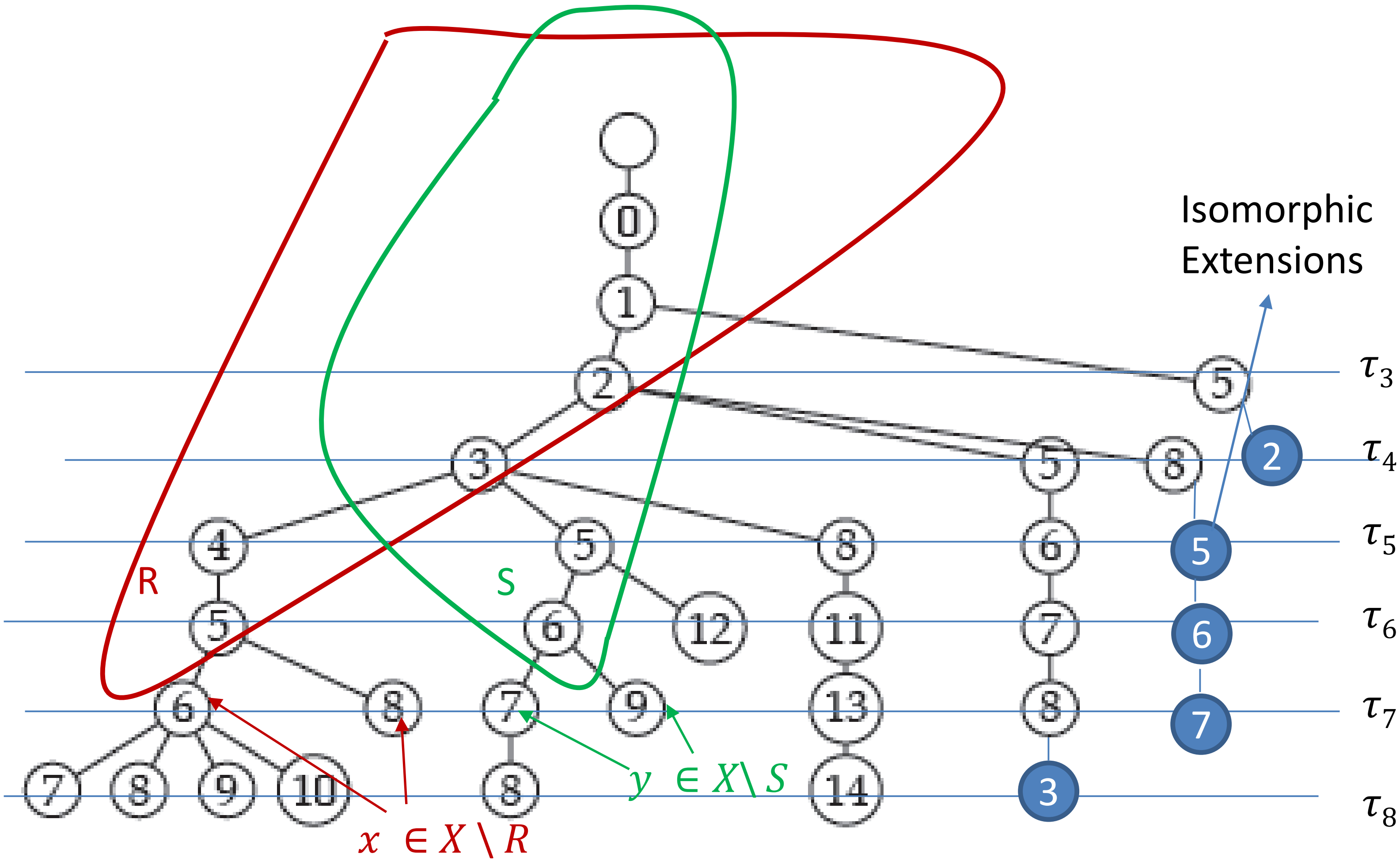
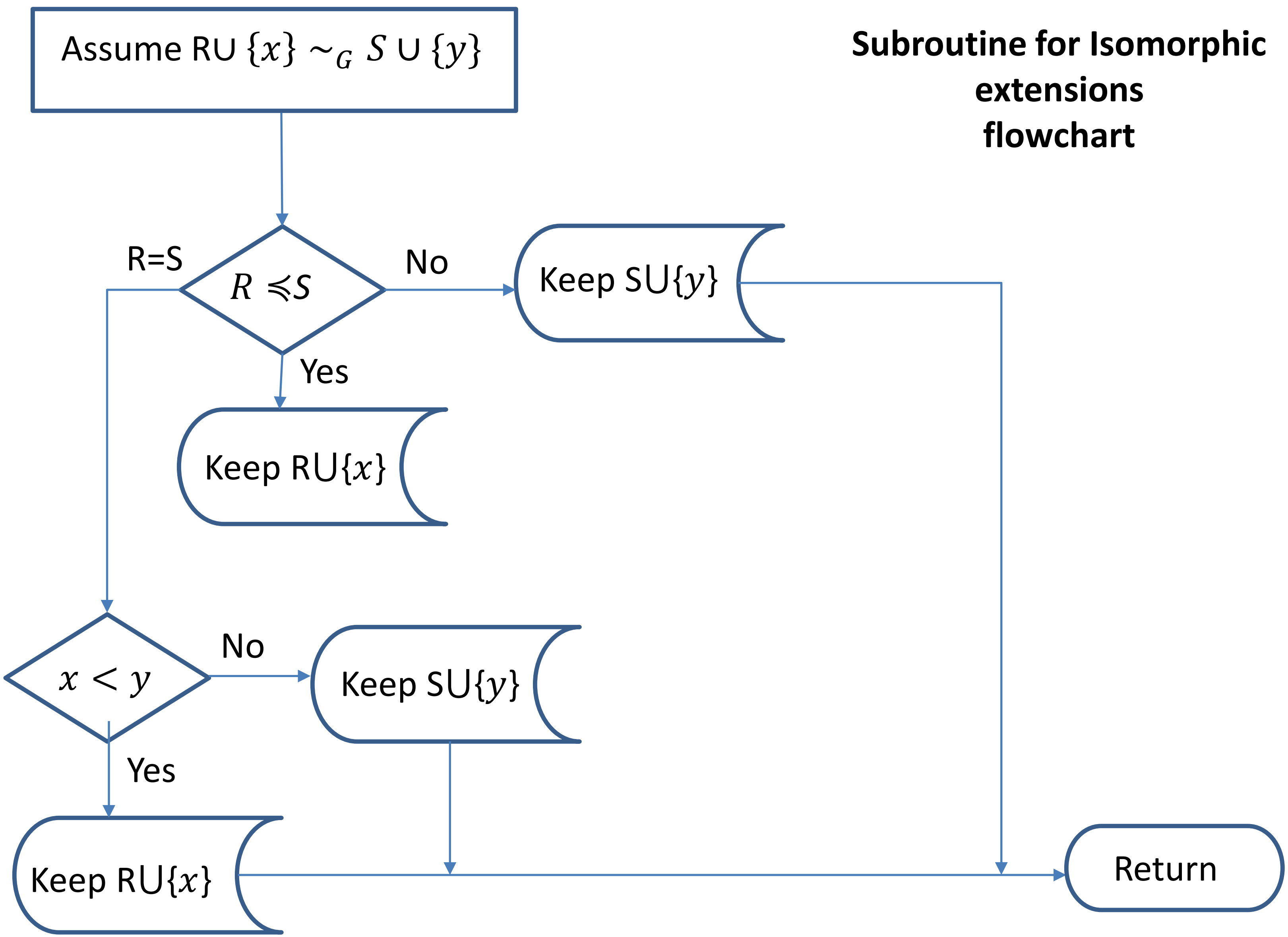


Fig. 9.4 Orbits of $PGL_4(2)$ on $\mathcal{P}_{\leq 8}(PG_3(2))$


Subroutine for Isomorphic extensions flowchart
Assume R∪ $\{x\} \sim_G S \cup \{y\}$
R=S
$R \preccurlyeq S$
No
Keep S∪$\{y\}$
Yes
Keep R∪$\{x\}$
$x < y$
No
Keep S∪$\{y\}$
Yes
Keep R∪$\{x\}$
Return

Problem 1 Redefinition

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Results got in problem 2 let us refine condition for Problem 1.

Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.

Corollary

It suffices to consider only extensions of the form

R$\cup$ $\{x\}$, where R is a $canonical\ i - set$ and

$x \in X \backslash R$ is $canonical$ under the $stabilizer$

G_R of R in G

Outline Presentation

- Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)
- Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.
- Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)

✓ **Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.**

- Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$
- General Algorithm of snakes and ladders for generation of codes.

Problem 3 Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 3: Compute Stabilizer in G of an extension $R \cup \{x\}$,
assuming that G_{R} is known.
If the extension is $R \cup \{x\}$, compute $G_{\mathrm{R} \cup \{x\}}$, a set-wise stabilizer.

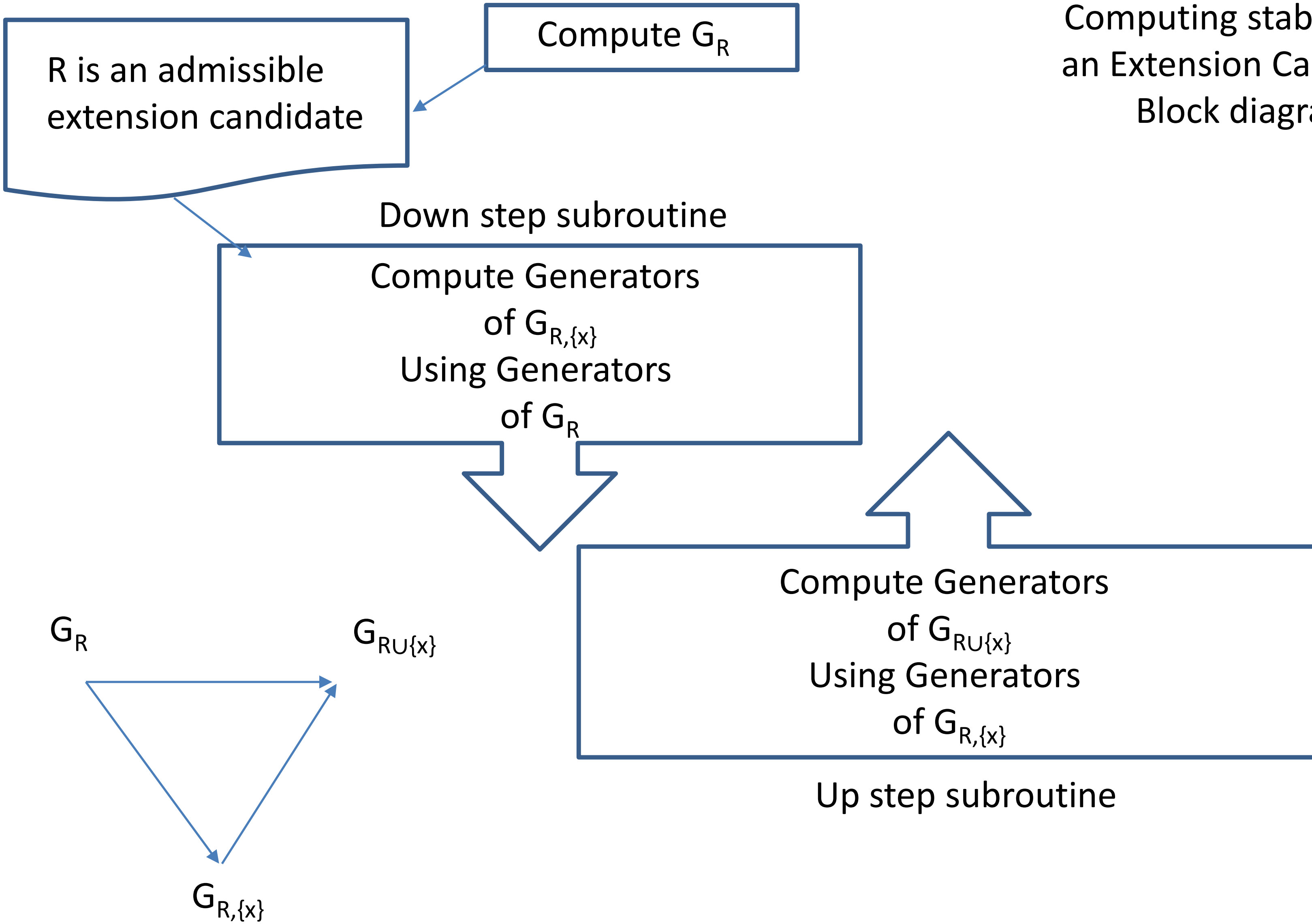
$\nexists$ relationship between groups G_{R} and $G_{\mathrm{R} \cup \{x\}}$,
neither is a subgroup of the other.

They share a common subgroup, $G_{\mathrm{R},x}$, set of elements which
stabilize R setwise and x pointwise.

First, go down from G_{R} to $G_{\mathrm{R},x}$ (**down step**, generators of $G_{\mathrm{R},x}$ can be computed from generators of G_{R} , using Corollary of Schreier Th.)

Second , to compute $G_{\mathrm{R} \cup \{x\}}$ from $G_{\mathrm{R},x}$ (**up step,** using some Lemmas)

Problem 3
Computing stabilizer of
an Extension Candidate
Block diagram



Theorem:(Otto Schreier)

$\exists\ \overline{rs} \in R \mid rs \in H\overline{rs}$.

Let G be a finite group generated by a set of elements S.

Let $H \leq G$.

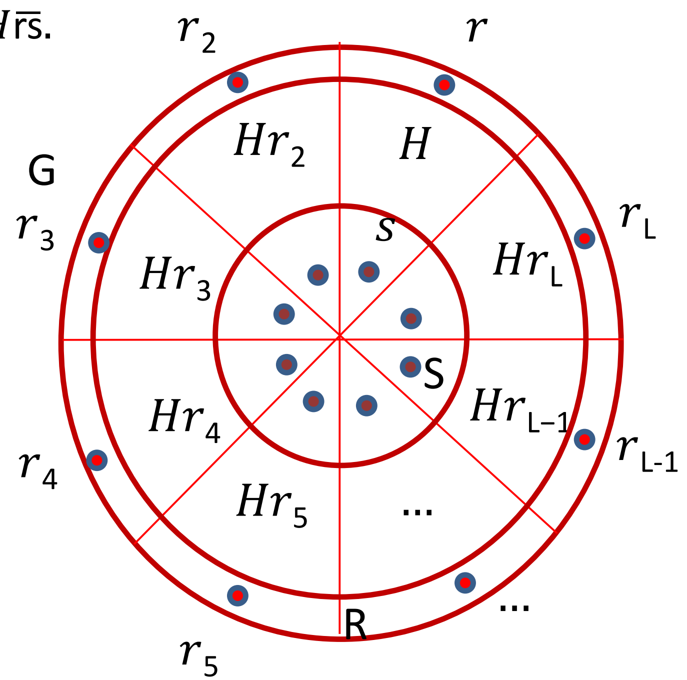
Given s $\in G$ Let R=$\{r \in G \mid Hr \subseteq G/H\}$ containing 1.

For $r \in R, s \in S$, Let $\overline{rs}$ be the unique element in R with $rs \in H\overline{rs}$.

Then,

$$H = \langle rs\,\overline{rs}^{-1} \rangle \quad , r \in R\ , s \in S$$

$H = \langle rs\,\overline{rs}^{-1} \rangle$



Each $\overline{rs}$ is called **Schreier generator.**

Remarks: Recall that $G = \cup_{r \in R} Hr$

Notice that $\overline{g} \in R$ the unique element s.t. $g \in H\,\overline{g}$

then $\overline{hg} = \overline{g}$ if h$\in H$

$\overline{\overline{g}} = \overline{g} \quad \forall\, g \in$ G

$\overline{g} = 1\ \ if\ g \in$ H

Corollary of Scheirer Th.

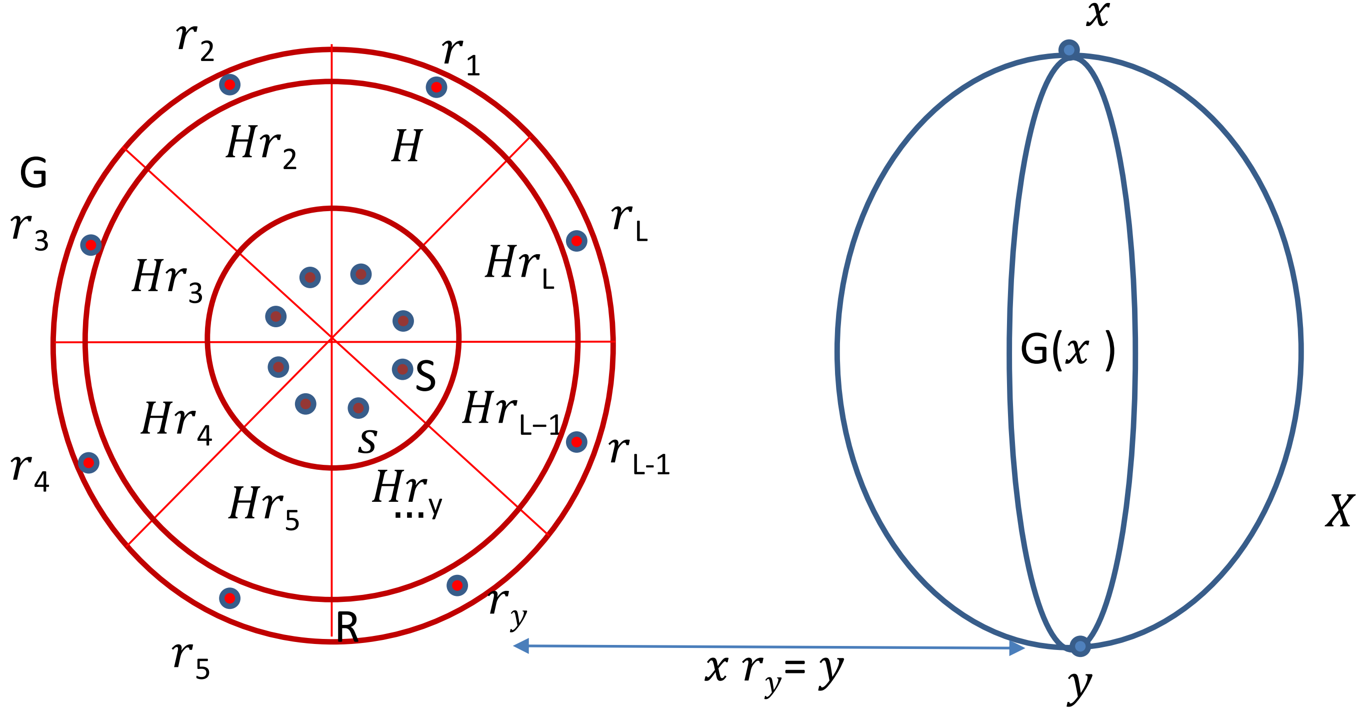
Let group G act on a finite set X and let S be a set of generators for G.

For $x \in X$, let R=$\{r_1, r_2,\ldots r_{\mathrm{L}}\}$ with r_1=1 set of elements s.t. :

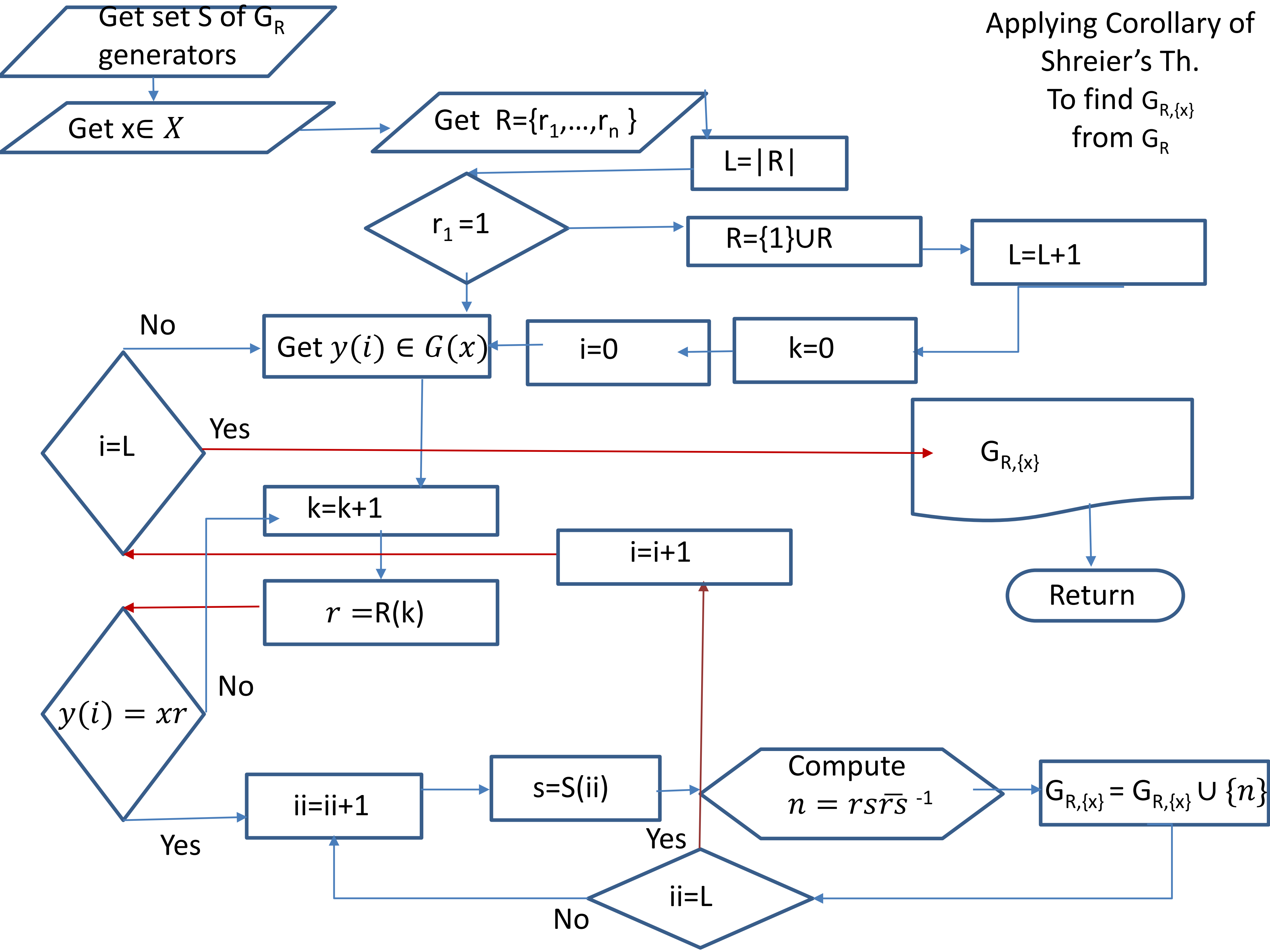
For each $y \in G(x)$ $\exists$ one and only one $r_y \in$R with x$r_y = y$, ($|G(x)| = \mathrm{L}$)

Then

$G_x = \{$s $\in G \mid$ s$x = x\}$ $\quad$ $G_{\mathrm{x}} = \langle r_y s \overline{r_y s}^{-1} \rangle$ $\quad$ $r_y \in$ R, $s \in$ S



$\exists\ \overline{r_y s} \in R \mid\ r_y s \in H\overline{r_y s}$



Applying Corollary of Shreier's Th.
To find $G_{R,\{x\}}$ from G_R
Get set S of G_R generators
Get x∈ X
Get R={r_1,…,r_n }
L=|R|
r_1 =1
R={1}UR
L=L+1
No
Get $y(i) \in G(x)$
i=0
k=0
i=L
Yes
$G_{R,\{x\}}$
k=k+1
i=i+1
Return
r =R(k)
No
$y(i) = xr$
Compute
$n = rs\overline{rs}^{-1}$
$G_{R,\{x\}}$ = $G_{R,\{x\}}$ U $\{n\}$
ii=ii+1
s=S(ii)
Yes
Yes
ii=L
No

Problem 3 Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 3: Compute Stabilizer in G of an extension $R \cup \{x\}$, assuming that G_R is known.

Lemma :

For an action ${}_G X$ the following holds:
$\forall x \in X$, the mapping
θ_x: $G_x / Gx \rightarrow G(x)$:
$g G_x \mapsto gx, g \in G$
Is a bijection between
The set
$G/ G_x = \{g\ G_x \mid g \in G\}$

For Finite sets $|G(x)|$ $|G: G_x| = |G/ G_x| = |G|/| G_x|$

Bijection Orbit Space and Cosets representatives of Stabilizer

Let S be a G-set, with $s \in S$ and G_s. $\forall\, g,h \in G$, $g \cdot s = h \cdot s \iff$ if $g\,G_s = h\,G_s$. $\therefore$, $\exists$ a bijection between elements of the orbit of s and cosets of the stabilizer G_s.

$$g \cdot s \leftrightarrow g\,G_s$$

$$h \cdot s \leftrightarrow h\,G_s$$

$$g \cdot s \leftrightarrow h\,G_s \iff g \cdot s = h \cdot s \iff g,h \in G(s)$$

$$h \cdot s \leftrightarrow g\,G_s \iff h \cdot s = g \cdot s \iff g,h \in G(s)$$

$\forall\ s \in {}_G S$, $G(s) \simeq$ coset action on G_s.

Bijection Orbit Space and Cosets representatives of Stabilizer

(Classification of G-Sets)

Let G be a finite group, and S a finite G-set. Then S is isomorphic to a union of coset actions of G on subgroups.

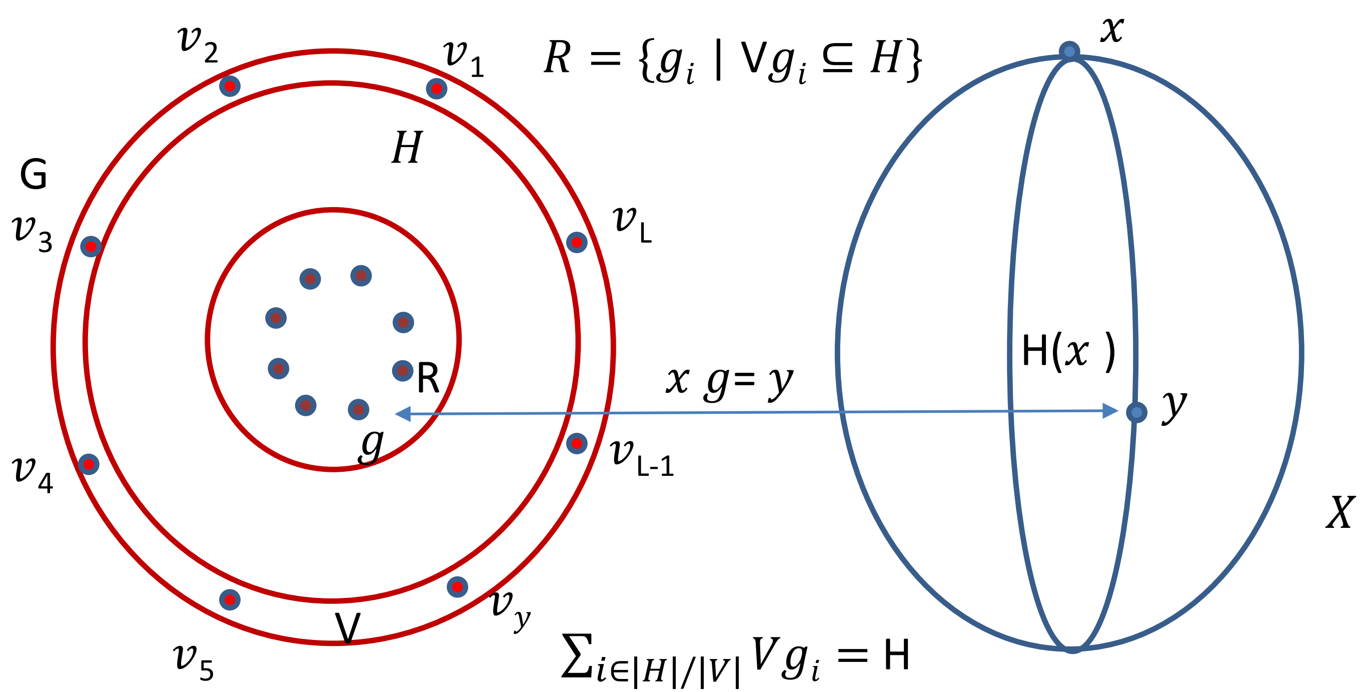
$$S = \cup_i G(si) \backsimeq \cup_i hG_{s_i}, \quad h \in G$$

Problem 3: Compute Stabilizer in G of an extension $R \cup \{x\}$, assuming that G_R is known.

Lemma :

Let H act on a set X , Let V=H_x be point-wise stabilizer of $x \in X$
Let R be a set of elements of H s.t. for each $y \in H(x)$, $\exists$ *one and only one* $g \in R$
With $xg = y$. Then R is a set of right coset representatives of V in H.

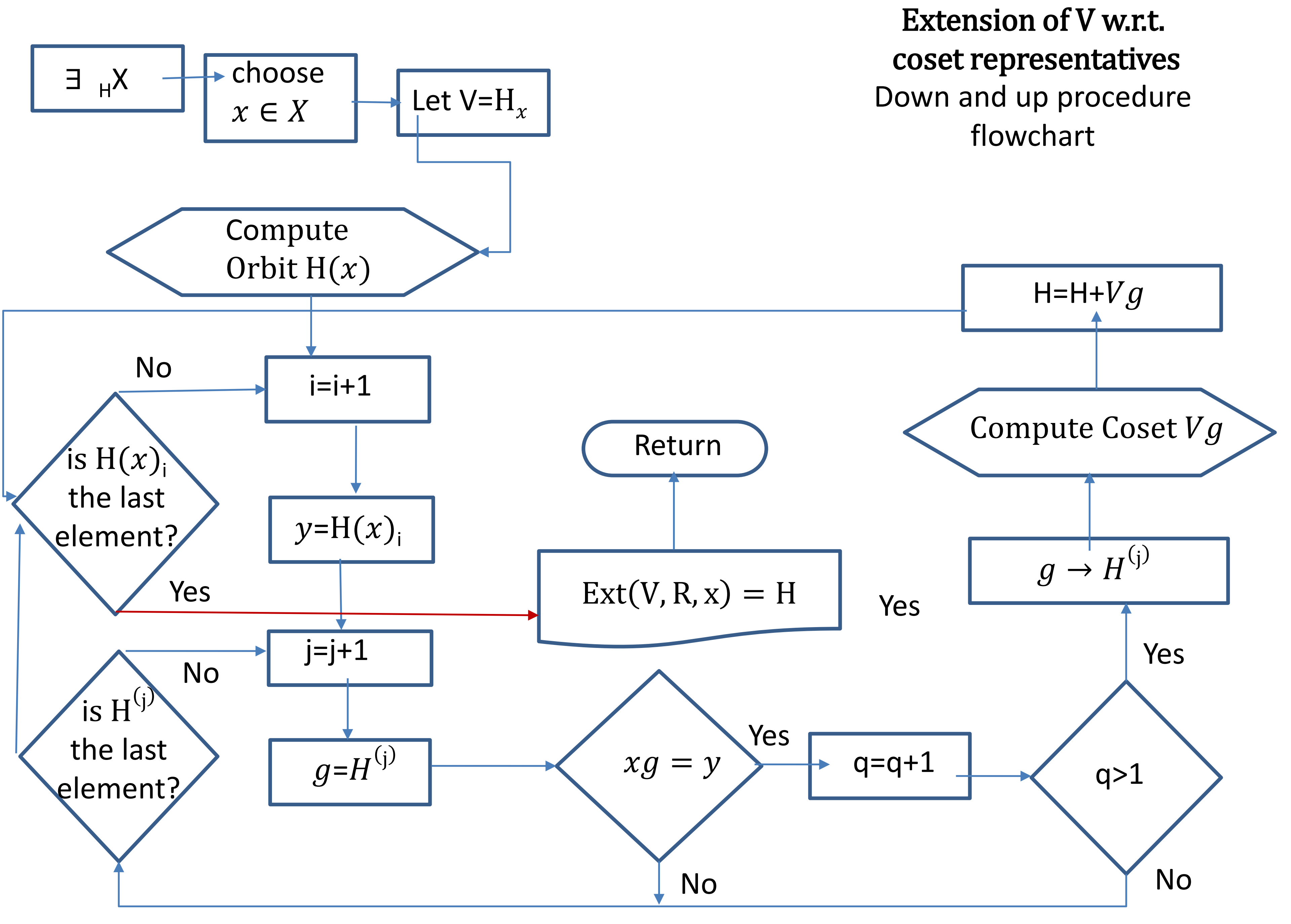
We call H the extension of V w.r.t. , the coset representatives R ,



Problem 3: Compute Stabilizer in G of an extension $R \cup \{x\}$, assuming that G_R is known.

Extension of V w.r.t. coset representatives

H= Ext (V, R, x) = $\cup_{r \in R} Vr$, union over disjoint cosets.


Extension of V w.r.t.
coset representatives
Down and up procedure
flowchart
∃ $_H$X
choose
$x \in X$
Let V=H_x
Compute
Orbit $H(x)$
H=H+Vg
No
i=i+1
is $H(x)_i$
the last
element?
Return
Compute Coset Vg
y=$H(x)_i$
$g \to H^{(j)}$
Yes
Ext(V, R, x) = H
Yes
j=j+1
No
Yes
is $H^{(j)}$
the last
element?
g=$H^{(j)}$
$xg = y$
Yes
q=q+1
q>1
No
No

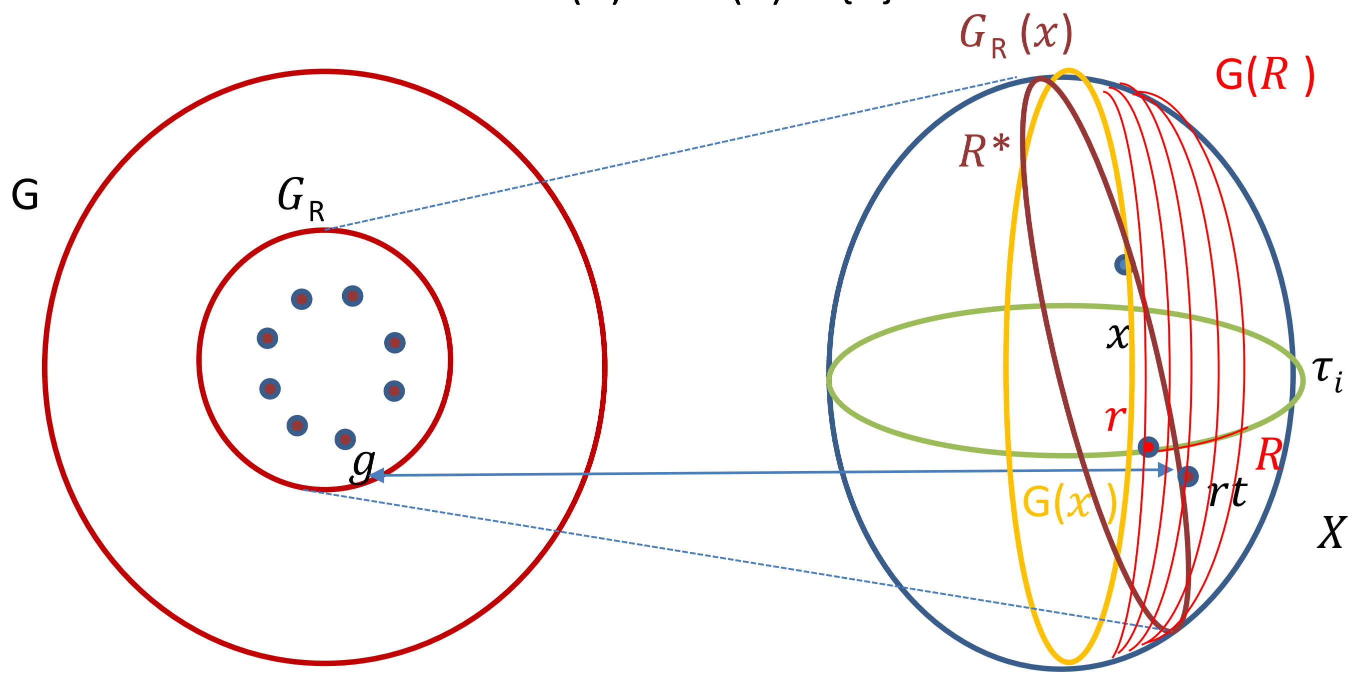
Problem 3: Compute Stabilizer in G of an extension $R \cup \{x\}$, assuming that G_R is known.

Special orbit sets R*(x) and R(x)

for $R \in \tau_i$, $x \in X \setminus R$,

$R^*(x) := \{r \in R \mid (R\setminus\{r\}) \cup \{x\} \in G(R)$ and $rt \in G_R(x)\}$ where $t = \varphi_i((R\setminus\{r\}) \cup \{x\})$.

Let $R(x) := R^*(x) \cup \{x\}$



Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 3: Compute Stabilizer in G of an extension R ∪ $\{x\}$, assuming that G_{R} is known.

Lemma

Let G act on a set X ,

Assume $(G, \wp_i(\mathrm{X}))= (\tau_i, \sigma_i, \varphi_i)$, for $R \in \tau_i$,

Let $(G_R, X \setminus R)= (\tau_R, \sigma_R, \varphi_R)$,

fix $x \in \tau_R$,

Then $R(x)= (G_{\mathrm{R} \cup \{x\},} x)$

In particular $\mathbf{G_{R \cup \{x\}}}$= **Ext** $(\boldsymbol{G_{\mathrm{R},x}}, \boldsymbol{R}, \boldsymbol{x})$

where

R=$\{1\}$∪ $\{t. \varphi_r(rt) | r \in R*(x)$,

with $t = \varphi_{\mathrm{i}\,((} R\setminus\{r\})$∪ $\{x\}$)}.

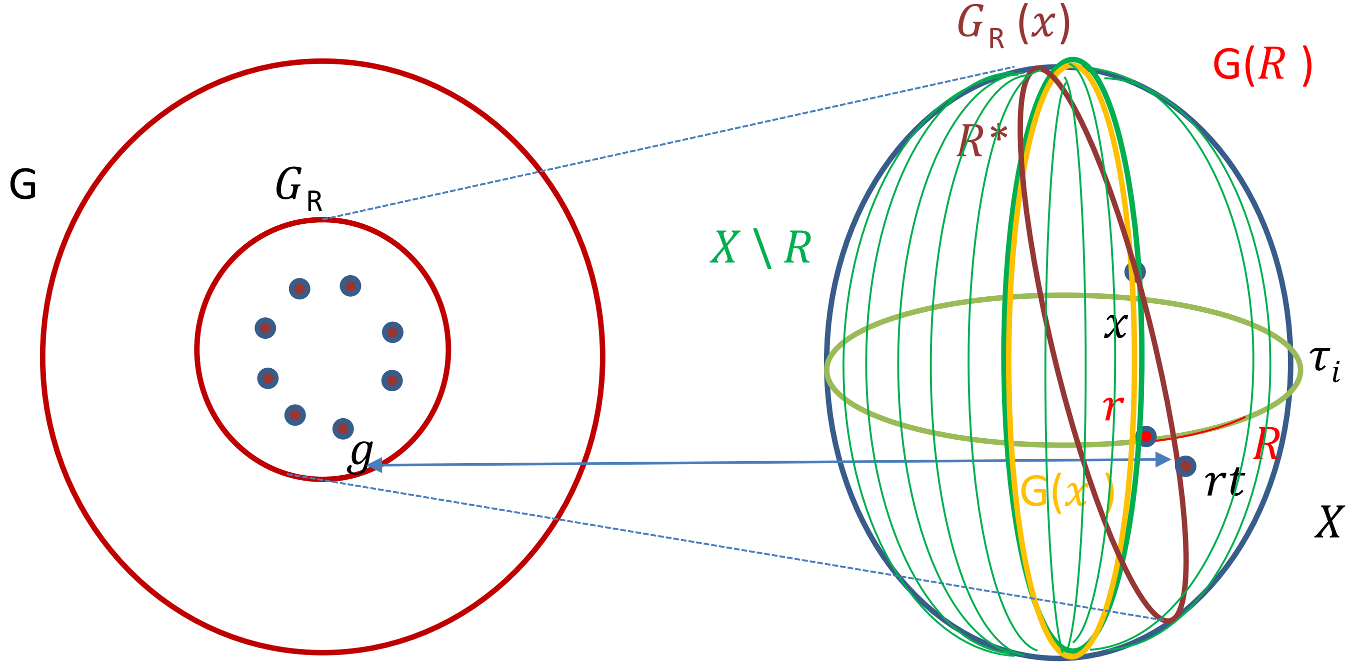
Here $\boldsymbol{G_{\mathrm{R},x}}$ = $G_{\mathrm{R}} \cap G_x$ = $\{g \in G | Rg = R, xg = x\}$

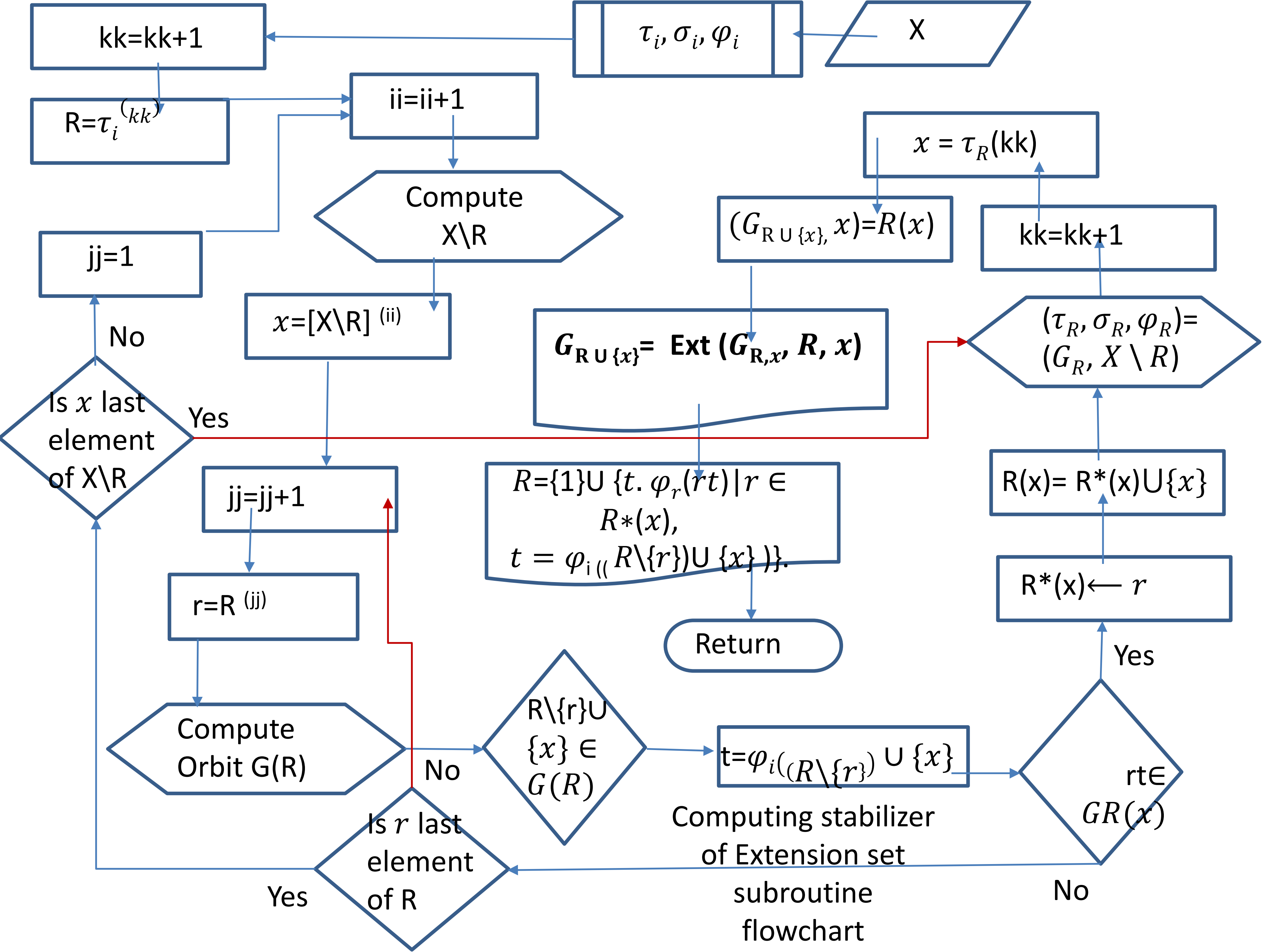
and

$G_{\mathrm{R} \cup \{x\}}$ is the **set-wise stabilize**r of the set **R ∪ $\{x\}$**

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 3: Compute Stabilizer in G of an extension R U $\{x\}$, assuming that G_R is known.





kk=kk+1
τi, σi, φi
X
R=τi(kk)
ii=ii+1
x = τR(kk)
Compute X\R
(GR∪{x}, x)=R(x)
kk=kk+1
jj=1
x=[X\R](ii)
GR∪{x}= Ext (GR,x, R, x)
(τR, σR, φR)= (GR, X \ R)
No
Is x last element of X\R
Yes
jj=jj+1
R={1}∪ {t. φr(rt)|r ∈ R∗(x), t = φi((R\{r})∪ {x})}.
R(x)= R*(x)∪{x}
r=R(jj)
R*(x)⟵ r
Return
Yes
Compute Orbit G(R)
No
R\{r}∪{x} ∈ G(R)
t=φi((R\{r}) ∪ {x}
rt∈ GR(x)
Is r last element of R
Computing stabilizer of Extension set subroutine flowchart
Yes
No

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 3: Compute Stabilizer in G of an extension R ∪ $\{x\}$, assuming that G_{R} is known.

Lemma

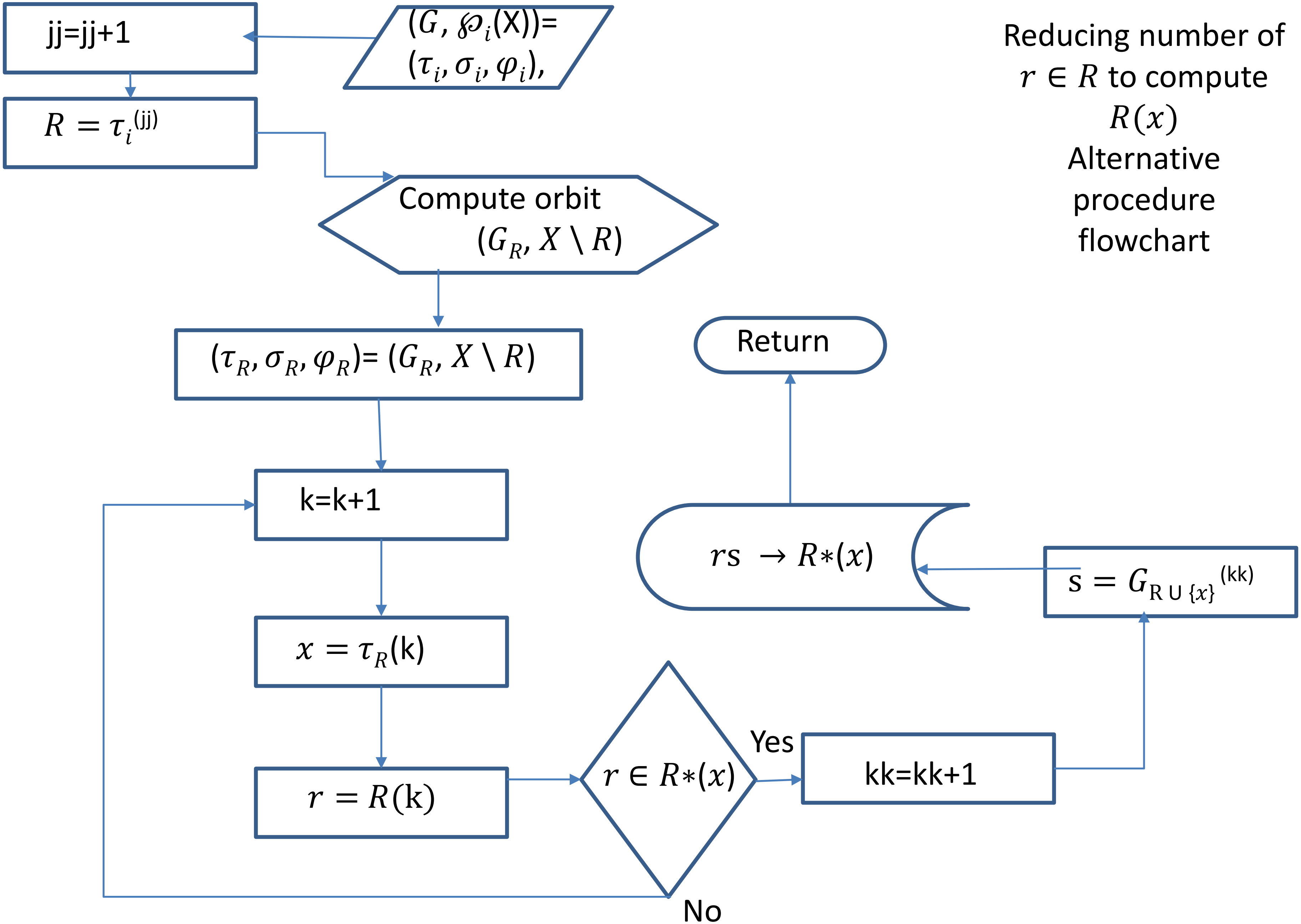
Let the group G act on a set X , Assume that orbit $(G, \wp_i(\mathrm{X}))= (\tau_i, \sigma_i, \varphi_i)$, for $R \in \tau_i$, Let orbit $(G_R, X \setminus R)= (\tau_R, \sigma_R, \varphi_R)$, fix $x \in \tau_R, r \in R$

Then,

1. if $r \in R*(x)$ then $rs \in R*(x)$ for all s $\in G_{\mathrm{R} \cup \{x\}}$.
2. if $r \notin R*(x)$ then $rs \notin R*(x)$ for all s $\in G_{\mathrm{R} \cup \{x\}}$.

Reducing number of $r \in R$ to compute $R(x)$

Alternative procedure flowchart



Outline Presentation

- Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)
- Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.
- Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)
- Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.
- ✓ **Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$**
- General Algorithm of snakes and ladders for generation of codes.

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$

Find $g = \varphi_{i+1}(H)$

s.t. $\mathrm{H}g = R$, H$\in G(R)$

Given a set F of size i+1 , the question is to find the canonical representative

$R \cup \{x\} \in \tau_{i+1}$, with $F \sim_G R \cup \{x\}$.

We wish to determine an element $g \in G$ with $Fg = R \cup \{x\}$.

The problem is solved recursively,

F is split into z:=max F and H= F \{z} .

By induction we can compute an element $t = \varphi_i$ (H),

Then S :=Ht is a canonical orbit representative

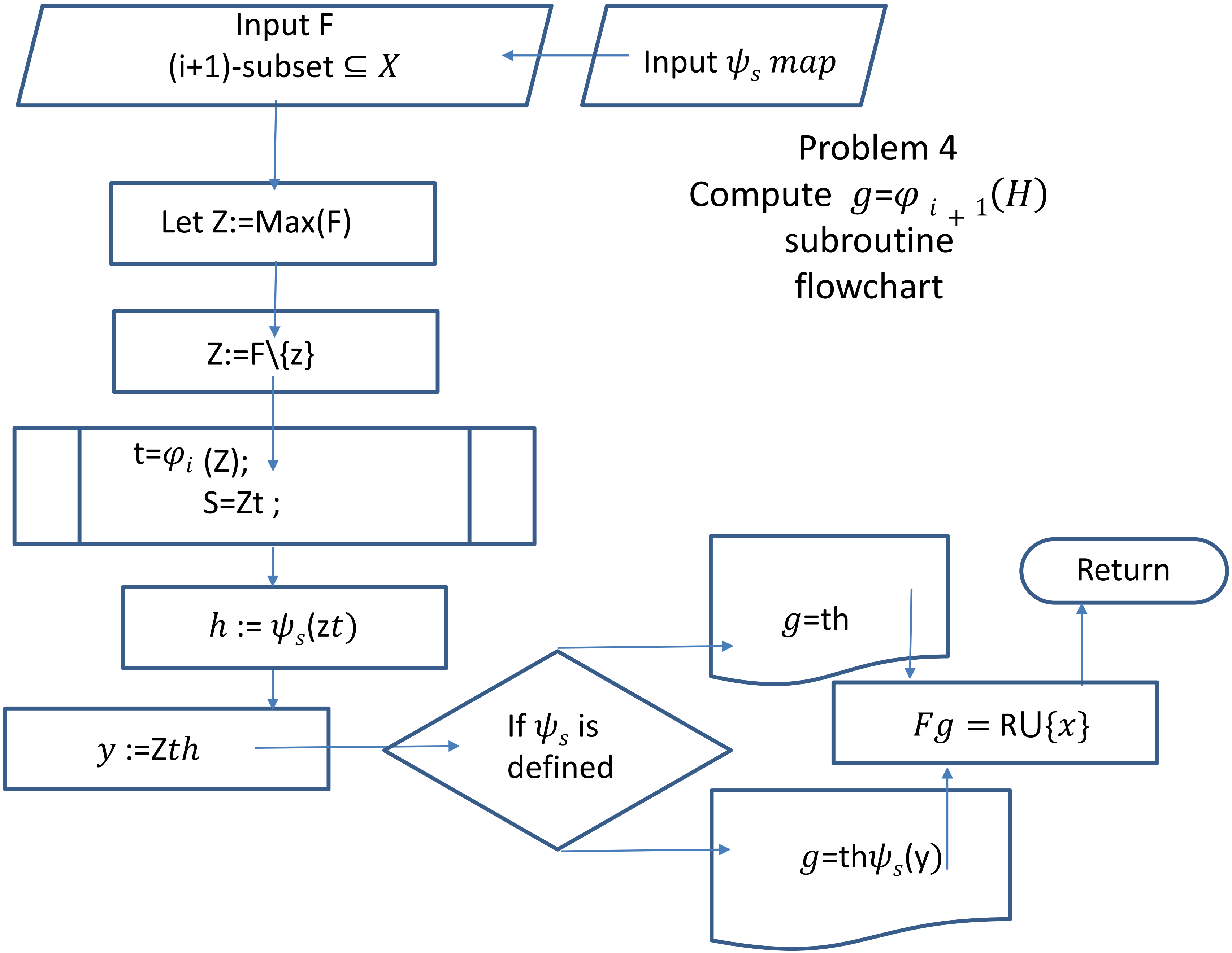
Using orbit data compute $h \in Gs$ s.t. $hth = y$ is canonical under G_s.

If $S \cup \{y\}$ is canonical under G, we return th,

Otherwise, if $S \cup \{y\}$ is a fusion node , then

We have a fusion element $\psi s(y)$, $s.t.$

$(S \cup \{y\})\, \psi s(y) = R \cup \{x\}$ is canonical


Input F
(i+1)-subset ⊆ X
Input ψ_s map
Problem 4
Compute $g=\varphi_{i+1}(H)$
subroutine
flowchart
Let Z:=Max(F)
Z:=F\{z}
t=φ_i (Z);
S=Zt ;
h := ψ_s(zt)
y :=Zth
If ψ_s is
defined
g=th
g=thψ_s(y)
Fg = RU{x}
Return

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Where the function φ_{i+1} is defined as follows.

function $\varphi_{i+1}(F)$
    $z := \max F, Z := F \setminus \{z\}$ (a set of size i)
    $t := \varphi_i(Z)$
    $S := Zt$
    $h := \varphi_S(zt), y := zth$
    if $\psi_S(y)$ has been defined **then**
        return $th\psi_S(y)$
    else
        return th
    end if
end function

Outline Presentation

- Construction of Codes when we can't compute a canonical from every subset (Snakes and Ladders Algorithm-Overview)
- Problem 1: Ensure that each G -orbit on admissible (i+1)–sets is reached.
- Problem 2: Determine when 2 extensions are isomorphic $R \cup \{x\}$ and $S \cup \{y\}$, i.e. belong to the same G -orbit. (R and S are both canonical, since $R, S \in \tau_i$)
- Problem 3: Compute in y the Stabilizer $G_{R \cup \{x\}}$, assuming that G_R is known.
- Problem 4: Provide a transporter map φ_{i+1} for (i+1)-sets, that is, given a (i+1)-subset $F \subseteq X$, compute $g \in G$ s.t. $Fg \in \tau_{i+1}$

✓ **General Algorithm of snakes and ladders for generation of codes.**

Snakes and ladders algorithm (Leiterspiel , B. Schmalz, 1992)

Theorem *Let G act on the finite set X. Assume that we can compute stabilizers, group extensions and orbits on points for subgroups of G. Furthermore, let* $f : \mathcal{P}(X) \to \{0,1\}$ *be a test function which is G-invariant and hereditary (in the sense of 9.5.1 and 9.5.2). Then Algorithm 9.6.10 computes the orbits of G on* $\mathcal{P}^{(f)}(X) = \mathcal{P}(X) \cap f^{-1}(\{1\})$*, the set of admissible subsets of X.* □

$$f \text{ is} \begin{cases} 1 \;\; iff \quad S \text{ is admissible} \\ f(S) = f(Sg) \;\; \forall g \in G, \forall S \subseteq X & \text{invariant under action of a group} \\ f(S) = 1 \;\Rightarrow f(T) = 1, \forall\, T \subseteq S \subseteq X & \text{Hereditary property} \end{cases}$$

Algorithm (orbits on subsets)

Input: $\text{orbit}\big(G, \mathcal{P}_i^{(f)}(X)\big) = (\mathcal{T}_i, \sigma_i, \varphi_i)$

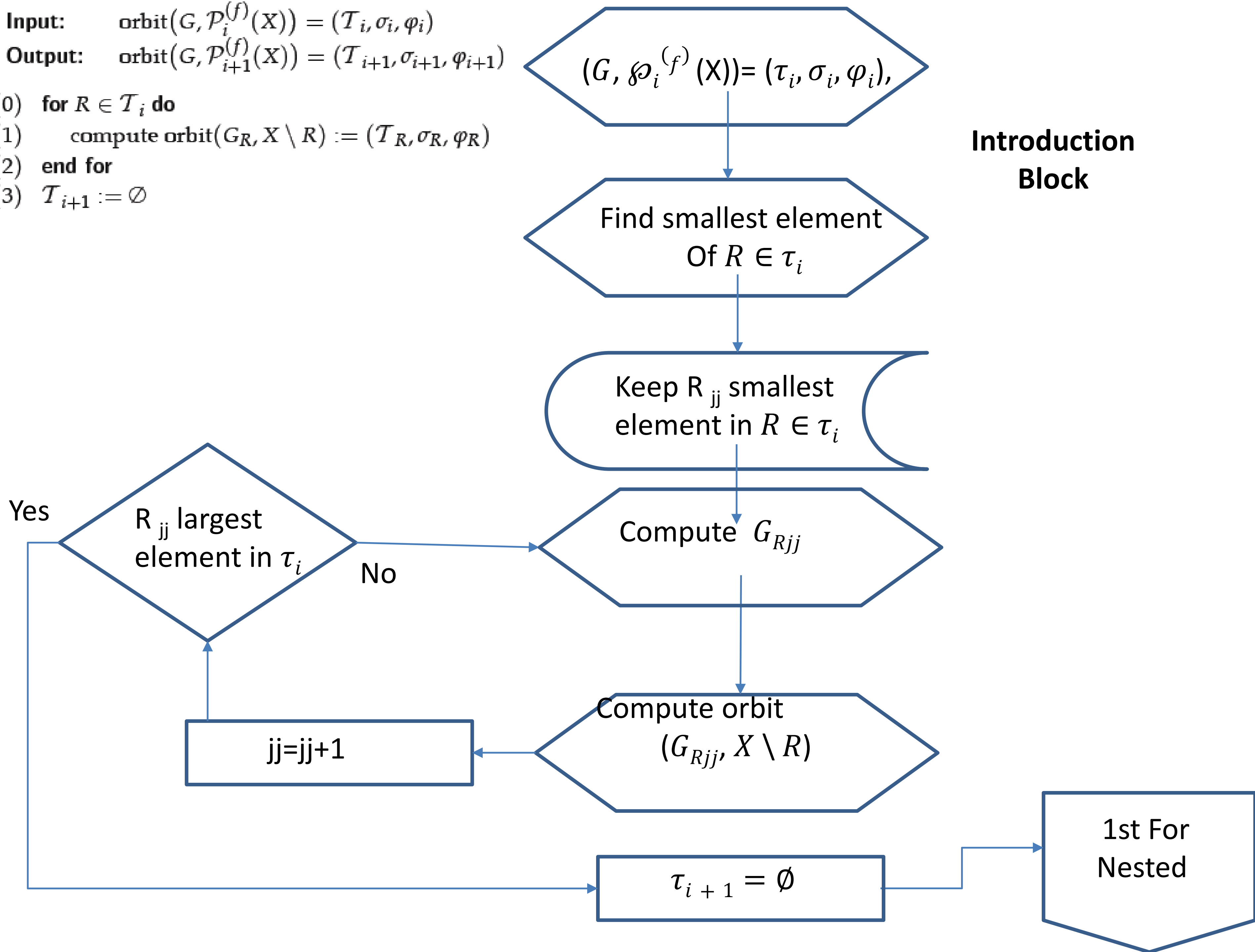
Output: $\text{orbit}\big(G, \mathcal{P}_{i+1}^{(f)}(X)\big) = (\mathcal{T}_{i+1}, \sigma_{i+1}, \varphi_{i+1})$

(0) **for** $R \in \mathcal{T}_i$ **do**
(1)   compute $\text{orbit}(G_R, X \setminus R) := (\mathcal{T}_R, \sigma_R, \varphi_R)$
(2) **end for**
(3) $\mathcal{T}_{i+1} := \emptyset$
(4) **for** $R \in \mathcal{T}_i$ (in increasing order) **do**
(5)   **for** $x \in \mathcal{T}_R$ (in increasing order) with $f(R \cup \{x\}) = 1$
            and for which $\psi_R(x)$ has not yet been defined **do**
(6) $G_{R,x} := \sigma_R(x)$
(7) $H := G_{R,x}$
(8)   **for** all $r \in R$ which are least in their H-orbit **do**
(9) $t := \varphi_i((R \setminus \{r\}) \cup \{x\})$
(10) $S := ((R \setminus \{r\}) \cup \{x\})t$
(11) $h := \varphi_S(rt)$
(12) $y := rth$
                (now: $(R \cup \{x\})th = S \cup \{y\},\ S \in \mathcal{T}_i,\ y \in \mathcal{T}_S$)
(13)   **if** $S = R$ **and** $y = x$ **then** (case 1 of 9.6.1)
(14) $H := \langle H, th \rangle$
                    (th is an automorphism of $R \cup \{x\}$)
(15)   **else** (case 2 of 9.6.1)
(16) $\psi_S(y) := (th)^{-1}$
                    (th is an isomorphism from $R \cup \{x\}$ to $S \cup \{y\}$)
(17)   **end if**
(18)   **end for**
(19)   append $R \cup \{x\}$ to $\mathcal{T}_{i+1}$
(20) $\sigma_{i+1}(R \cup \{x\}) := H \ (= G_{R \cup \{x\}})$
(21)   **end for**
(22) **end for**
(23) **return** $(\mathcal{T}_{i+1}, \sigma_{i+1}, \varphi_{i+1})$

Input: $\text{orbit}(G, \mathcal{P}_i^{(f)}(X)) = (\mathcal{T}_i, \sigma_i, \varphi_i)$

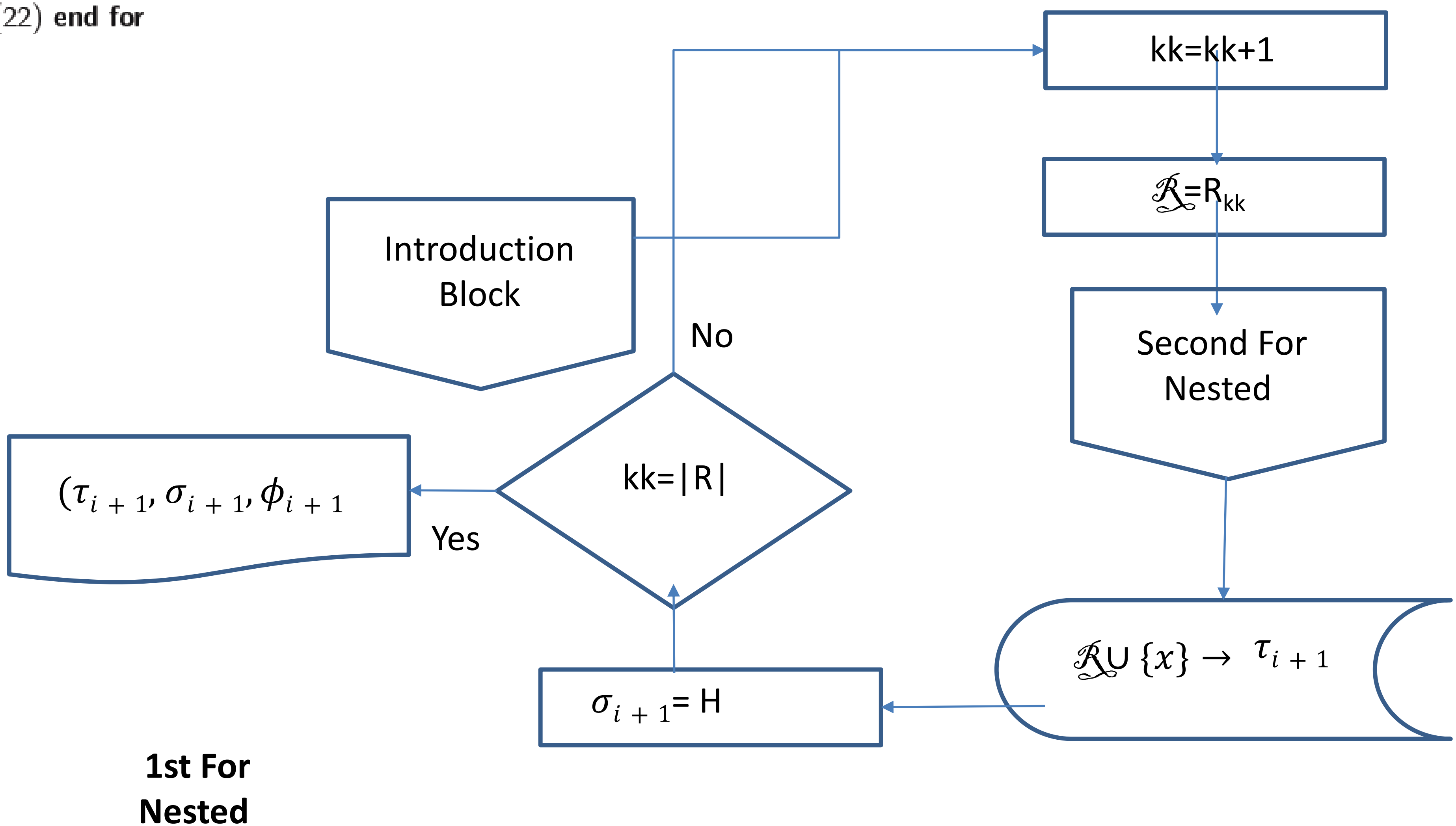
Output: $\text{orbit}(G, \mathcal{P}_{i+1}^{(f)}(X)) = (\mathcal{T}_{i+1}, \sigma_{i+1}, \varphi_{i+1})$

for $R \in \mathcal{T}_i$ **do**
  compute $\text{orbit}(G_R, X \setminus R) := (\mathcal{T}_R, \sigma_R, \varphi_R)$
end for
$\mathcal{T}_{i+1} := \emptyset$



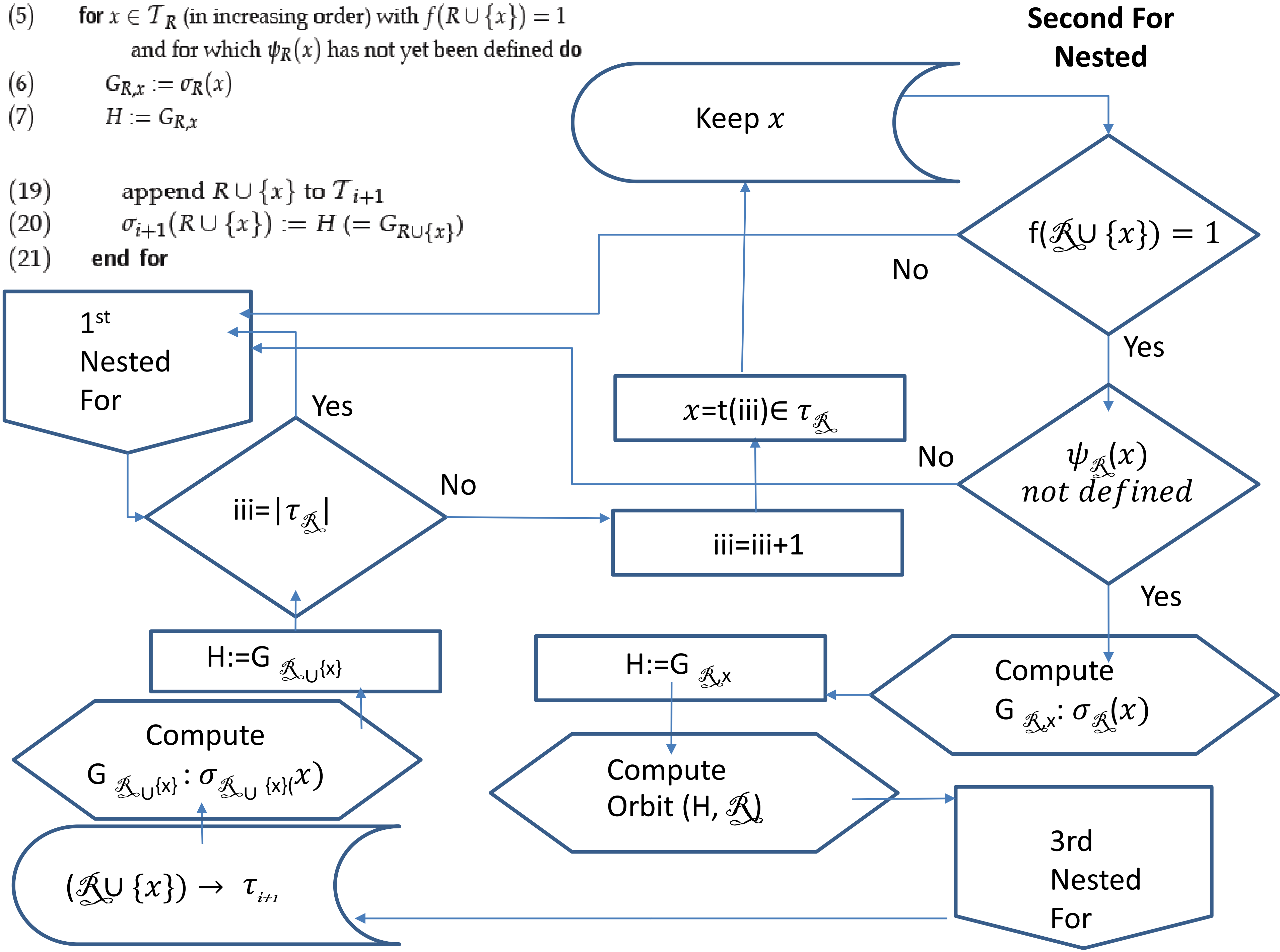
(4) **for** $R \in \mathcal{T}_i$ (in increasing order) **do**

(22) **end for**



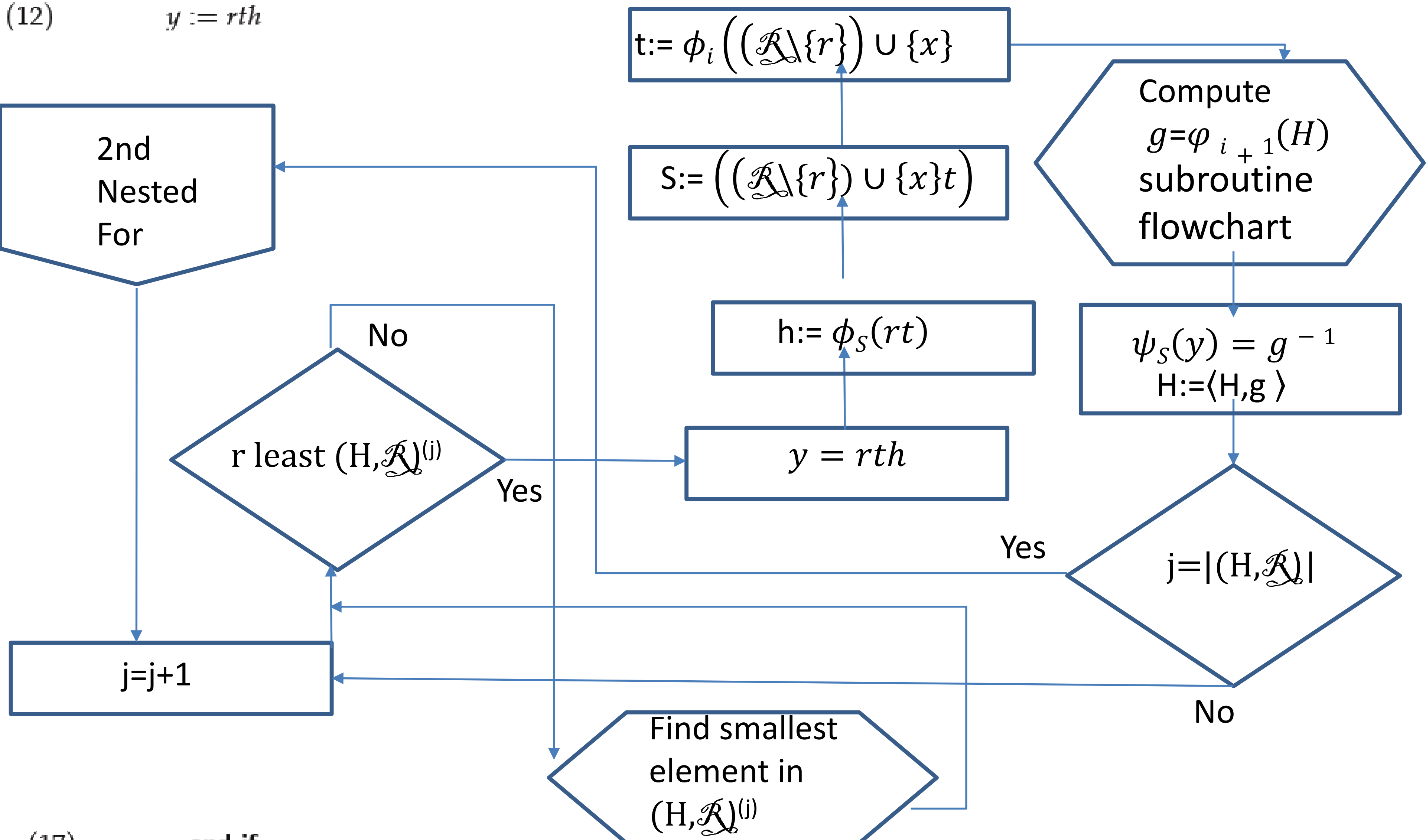
```
(5)   for x ∈ T_R (in increasing order) with f(R ∪ {x}) = 1
          and for which ψ_R(x) has not yet been defined do
(6)       G_{R,x} := σ_R(x)
(7)       H := G_{R,x}

(19)      append R ∪ {x} to T_{i+1}
(20)      σ_{i+1}(R ∪ {x}) := H (= G_{R∪{x}})
(21)  end for
```



```
for all r ∈ R which are least in their H-orbit do
    t := φ_i((R \ {r}) ∪ {x})
    S := ((R \ {r}) ∪ {x})t
    h := φ_S(rt)
    y := rth
```

3rd Nested For




```
        end if
    end for
```